תוכן מסמך זה הוא דוגמא טובה לספר פרויקט, תמיד יש מה לתקן.

יש להפעיל שיקול דעת ולהתאים ספציפית לפרויקט בהתאם להנחיות הניתנות בנוהל הפרויקטים העדכני לאותה שנה ובהתאם להנחיות המנחה של הפרויקט.

מחלקת הנדסת תוכנה

שם הפרויקט:

מערכת מידע לניהול, תיעוד ורישום אירועי אבטחת מידע עבור מוקד אבטחת מידע של בנק דיסקונט

ספר פרויקט

שם הסטודנט:	ויקטור קייטמזוב
מספר תעודת זהות:	306859893
שם המנחה:	גדעון קוך
תאריך ההגשה:	15.05.08

1. פתיחה (Introduction)

a. תקציר מנהלים

מטרת ספר הפרויקט הינה להציג בצורה מקיפה את פרויקט הגמר- מערכת מידע לניהול, תיעוד ורישום אירועי אבטחת מידע עבור מוקד אבטחת מידע של בנק דיסקונט.

המטרות העיקריות של מערכת המידע היא ריכוז וסיווג כל האירועים החריגים שמתרחשים בזמן אמת תחת מערכת אחת. פיתוח מאגר ידע שיכיל את כל נהלי העבודה במוקד, מתן שליטה ובקרה על אירועים חריגים שנרשמים במערכת תוך כדי מתן אפשרות לנהל מעקב של טיפול בכל אירוע ודיווח לגורמים הנדרשים לפי נוהל העבודה. כמו כן המערכת תאפשר לנתח את התפלגויות השונות בין האירועים המתרחשים בזמן אמת באמצעות תצוגה גראפית ותנהל את הפקת הדו"חות בצורה יעילה.

בנק דיסקונט מנהל את המידע בצורה ממוחשבת באמצעות מערכות מתקדמות שמטפלות בנתוני הלקוחות של הבנק. מחלקת אבטחת מידע עובדת עם תוכנות מתקדמות שמאפשרות ניטור מדיניות אבטחת מידע של הבנק. הגוף שאחראי על ריכוז האירועים הוא מוקד אבטחת מידע שמאויש ע"י בקרי אבטחת מידע 24 שעות ביממה, 7 ימים בשבוע. חלק מהעבודה השוטפת של בקרי המוקד היא לזהות אירועים חריגים במערכות אבטחת המידע של הבנק ולדווח על כך לגורמים הנדרשים לפי נהלי העבודה של מחלקת אבטחת מידע. חשוב לציין שקיימות מספר מערכות שרצות במקביל ובכל מערכת מנטרת אירועים רלוונטיים המוגדרים בפרמטרים שלה, כלומר אין מערכת שתרכז את כל האירועים המתרחשים בזמן אמת ותאפשר מתן שליטה ובקרה באמצעות ניהול מעקב טיפול בכל האירועים ומכאן הצורך במערכת שפותחה במסגרת פרויקט הגמר.

מערכת המידע שפותחה בפרויקט באה ליישם את התפיסה החדשה של מערכות SIM(Security Information Management) ו SOC(Security operation Center) הקיימות היום בשוק. בעזרת מערכות SIM ניתן לזהות בכל רגע נתון את מצב אבטחת המידע בארגון, האיומים עמם הארגון מתמודד ודירוג חומרת הבעיות, ואילו SOC שהיא תמונה רחבה יותר של SIM מהווה ישות מרכזית לניהול ובקרה של אבטחת המידע בארגון.סקירה מפורטת של מערכות אלו ניתן לראות בסעיף g של פרק 1- סקירה ספרותית.

מערכת המידע כוללת 4 מודלים מרכזיים שבתוכם קיימים תתי מודולים נוספים שאחראיים לבצע את כל תהליכי המערכת ולבצע את כל הפונקציונאליות שהוגדרה בשלב אפיון הפרויקט. המודלים המרכזיים הם: ניהול אירועים (Event Manager), ניהול משתמשים (User Manager), ניהול ידע (Knowledge Manager) וניהול משמרות (Shift Manager).

הארכיטקטורה שבה נבנתה המערכת היא ארכיטקטורת השכבות המחולקת ל 4 שכבות: שכבה הפרזנטציה (Presentation Layer) האחראית על הצגת ממשק גראפי למשתמש, שכבת הלוגיקה (Business Logic Layer) האחראית על פעולות לוגיות של המערכת ומכילה מחלקות המייצגות את הישויות המרכזיות במערכת, שכבה קישור לבסיס הנתונים (Data Access Layer) האחראית על יצירת תקשורת בין שכבת הלוגיקה לשכבת בסיס הנתונים ושכבת בסיס הנתונים (Data Base) המכילה את נתוני המערכת בטבלאות בסיס הנתונים.

תהליך פיתוח המערכת מומש בסביבת Microsoft Visual Studio® .NET (2.0), כאשר התכנות עצמו היה בשפת #C ובסיס הנתונים שבו השתמשתי לפיתוח מערכת המידע הוא SQL Server שמובנה בתוך העורך.

במסגרת כל תהליך הפיתוח ואחריו התבצע בדיקות מערכת כפי שהוגדרו במסמך תכנון ועיצוב הבדיקות שמפורט בנספח ג - מסמך STD. פירוט תוצאות הבדיקות שבוצעו בהתאם להגדרתם במסמך STD ניתן לראות במסמך דווח בדיקות תוכנה בנספח ד' – מסמך STR.

כמו כן במשך כך תהליך פיתוח הפרויקט התבצע מעקב מתמיד של נושא ניהול וניתוח הסיכונים שהוא נושא מאוד חשוב להצלחת הפרויקט. סקירה יסודית ומפורטת של נושא ניתוח הסיכונים ניתן לראות בנספח ה - מסמך SPMP.

הבעיות העיקריות בתכנון הפרויקט היו איך לחלק את המודולים המרכזיים של הפרויקט בצורה הנכונה ביותר, מכיוון שתמיד נוצרים תתי מודולים תוך כדי שלב הפיתוח עצמו, ויש צורך להכניס שינויים שלא הוגדרו בתכנון הראשוני. זה נובע מחוסר ניסיון בתכנון ופיתוח מערכת בסדר גודל כזה.

כרגע ביצוע הפרויקט הושלם כאשר יש לקחת בחשבון שתמיד יש אפשרות לשפר דברים שכבר נכתבו ולכן כל עוד המוצר לא יסופק ללקוח, יתבצעו בדיקות יזומות לייעול תהליכי המערכת ומניעת תקלות במידת האפשר.

פיתוחים עתידיים שאפשריים למערכת הם התממשקות של המערכת למערכות הבנק על מנת לייעל את העבודה ולמנוע מבקרי המוקד עבודה מיותרת של הזנת נתוני האירועים למערכת, במידה וזה יתאפשר ניתן יהיה לבצע סינון אירועים אוטומטי לפי פרמטרים מוגדרים מראש, בצורה זו תופחת כמות האירועים שיגיעו למערכת ויינתן דגש רק לאירועים קריטיים שבהם צריך לטפל בזמן אמת.

The purpose of this document is to describe the final project - 7Tech - An information system to document and manage information security events for the information security center of bank discount

The main purpose of the 7Tech system is as follows:

a. Recording all unusual events at real time in a single system

b. Developing a knowledge base that will contain all the procedures in the information security center

c. Better control and tracking of unusual events

d. Reporting to the responsible according to the procedure

e. Better analysis of the various event categories, using graphical tools & reporting

The bank manages its information electronically, using many advanced systems which manage customer's data. The information security department utilizes advanced systems which monitor the various aspects of the bank's operational systems' activity, according to the bank's information security policy. The body responsible for this is the information security center, which is manned 24 hours a day, 7 days a week. Part of the daily routine is to identify unusual events and to report to the responsible departments, according to the policy. Currently, there are several systems involved in this process, each one in it's area, and there is no single system which consolidates all events – thus resulting in a need in such system.

The system developed in the project implements the new approach of SIM (Security Information Management) & SOC (Security Operations Center) systems that exists today. Using a SIM system, it is possible to monitor the current status of information security throughout the organization, including current threats and their severity levels. Further details regarding those two systems may be found in paragraph G of chapter 1.

The system is based on 4 primary modules, each containing several sub-modules. The main modules are:

a. Event Manager

b. User Manager

c. Knowledge Manager

d. Shift Manager

The system is built in n-tier architecture, composed of 4 tires:

a. The presentation layer – Responsible for the user interface
b. Business Logic layer – contains the primary objects and manages their relations
c. Data Access Layer – responsible for communication with the database
d. Database – responsible for storing the data

The system is developed in Microsoft Visual Studio .NET 2.0, using C#, with SQL Server as the database. During the development and afterwards, QA was performed as planned in the STD. The documentation of the tests performed can be found in the STR.

During the development a constant risk management process took place. Detailed information regarding this issue can be found in Appendix E – SPMP document.

The main challenge in this project was deciding how to break the program into modules in the most efficient way, keeping in mind that during the development process there are unexpected changes and additional sub-modules which need to be implemented. The main reason for this is my inexperience in planning and implementing software of this volume.

Currently, the project was completed according to the original design. It should be kept in mind that there is always an option to improve things as long as the product hasn't gone live.

In the future, it is possible to implement interfaces to the bank's operational systems. This will save to need to work with several systems in parallel and entering the data manually to the system. The interface will filter only certain types of events in the way that only critical security event will be focused in real time.

a. תוכן העניינים

פרק נושא **עמוד**

נספחים:

b. רשימות

c. מילון מונחים

להלן מילון מונחים כללי עבור הפרויקט:

DFD - Data Flow Diagram - תרשים זרימת נתונים
SQL - Structured Query Language - שפת שאילתות מובנית
GUI - Graphic User Interface - ממשק משתמש גראפי
SIM - Security Information Management - ניהול אבטחת מידע
SOC - Security operation Center - מוקד אבטחת מידע
DB - Data Base - בסיס נתונים
VB - Visual Basic - שפת תכנות

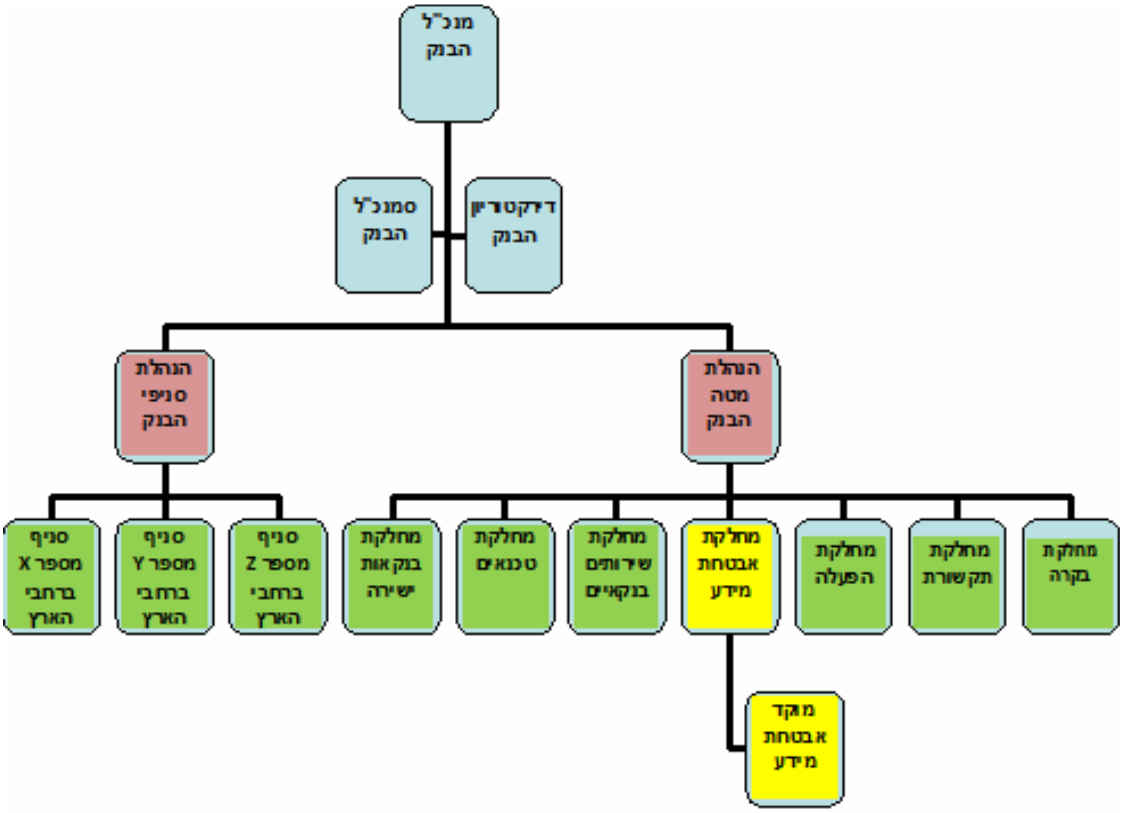
d. מבוא

הפרויקט הוא מערכת מידע לניהול, רישום ותיעוד אירועי אבטחת מידע. פרויקט זה הוא פרויקט אקדמי עם פוטנציאל תעשייתי.

המערכת מיועדת למחלקת אבטחת מידע של בנק דיסקונט. כחלק מהעבודה השוטפת קיים במחלקה מוקד אבטחת מידע שתפקידו לנטר את כל אירועי אבטחת המידע של הבנק. המוקד פעיל במשך 24 שעות ביממה, 7 ימים בשבוע ומאויש ע"י בקרי אבטחת מידע שעושים בקרה על מערכות הבנק. במוקד אבטחת מידע קיימות מספר מערכות שונות לניטור אירוע אבטחת מידע (שאותן אין באפשרותי לתאר עקב איסור מוחלט מטעם הבנק לפרסום מידע הקשור למערכות אלה).

כיום בעבודה השוטפת של הבנק בכלל ומחלקת אבטחת מידע בפרט קיימים עשרות סוגים של אירועים חריגים, כאשר לכל אירוע יש להכין דו"ח מיוחד ולהעבירו לגורמים הרלוונטיים בבנק. מערכת המידע שתבנה כחלק מהפרויקט תשמש את המוקד בעבודה השוטפת של תיעוד אירועים והפקת הדו"חות עבור אירועי אבטחת מידע חריגים, כמו כן המערכת תכיל נוהל טיפול עבור כל אירוע חריג שאירע באחת המערכות על מנת לאפשר מעקב יעיל של טיפול בכל אירוע חריג. בנוסף המערכת תאפשר הפקת דוחות פנימיים של המוקד, כגון: דו"ח חפיפה בחילוף המשמרות בין הבקרים במוקד ודו"ח מרוכז של אירועים חריגים לפי סינון רלוונטי עבור מנהל המוקד. דבר נוסף שיהיה במערכת הוא פורטל לניהול ידע שיכיל את כל הנהלים הרלוונטיים לעבודה השוטפת. חשוב לציין שבשלב הראשוני מערכת המידע שתבנה עבור המוקד לא תתממשק למערכות הבנק השונות אלא תשמש לניהול העבודה השוטפת של המוקד בלבד.

תרשים הארגון



איור 1 – תרשים הארגון

מבנה הארגון

מבנה הארגון מתחלק לשני גורמים מרכזיים: הנהלת מטה בנק והנהלת סניפי הבנק כאשר לכל גורם יש מחלקות משלו ורשת מחשבים משלו. הגורמים שרלוונטיים למערכת הם שני הגורמים המרכזיים (מסומנים בורוד) וחלק מהמחלקות הפנימיות של כל גורם (מסומנים בירוק). כאשר מתחיל טיפול באירוע חריג שנרשם במערכת, ברוב המקרים הגורמים הראשונים שלהם יש לדווח על האירוע החריג הם הגורמים המרכזיים שהם הנהלת מטה הבנק והנהלת סניפי הבנק (בהתאם למקור האירוע) ורק לאחר מכן בהתאם לסוג האירוע, הדיווח מגיע למחלקות הפנימיות לצורך המשך טיפול באירוע. בחלק קטן מהאירועים החריגים אין צורך להודיע לגורמים המרכזיים, אלא רק למחלקות הפנימיות של כל גורם.

הגורמים המעורבים בהכנת הפרויקט

לקוח מטעם הבנק: מאיר סלר - מנהל מוקד אבטחת מידע, בנק דיסקונט.

מנחה הפרויקט: גדעון קוך.

מפתח הפרויקט: ויקטור קייטמזוב.

e. מטרות ויעדי הפרויקט

מטרות העבודה, נשארו כפי שהוגדרו בפרק 3.3 של נספח ה' - הצעת פרויקט מורחבת והן:

- ריכוז כל האירועים החריגים מכל המערכות במוקד אבטחת מידע תחת מערכת אחת (פעולה זאת תעשה ע"י בקרי המוקד ולא ע"י התממשקות מערכת המידע למערכות הבנק השונות).
- הגדרת ופיתוח פורטל לניהול ידע שיכיל את כל נהלי העבודה עבור בקרי המוקד.
- ארגון מסודר של אירועים חריגים שנרשמו במערכת לפי סוג אירוע ורמת חומרתו.
- ניהול מסודר של דו"חות אירועים חריגים שאירעו במערכות השונות של המוקד.
- מתן שליטה ובקרה על אירועים חריגים תוך כדי אפשרות מעקב של טיפול בכל אירוע.

להסבר מפורט יותר ניתן לעיין בפרק 3 של נספח ד' – הצעת פרויקט מורחבת.

f. סקירה ספרותית

סקירה ספרותית, נשארה כפי שהוגדרו בפרק 3 של דו"ח ביניים 1 והיא:

מערכות SIM ו SOC – התפיסה החדשה של ניהול אבטחת המידע

SIM(Security Information Management)

למה צריך את SIM ?

מערכות המחשוב הארגוניות מייצרות כמות גדולה ומגוונת של התרעות. קשה להשתלט על שתף המידע המיוצר ע"י מערכות אלה, לזהות באמצעותו את האיומים העומדים בפני הארגון ולטפל בהם מבעוד מועד.

מערכות SIM - Security Information Management נועדו לעזור בהתמודדות עם אתגר זה. בעזרת מערכות אלו ניתן לזהות בכל רגע נתון את מצב אבטחת המידע בארגון, האיומים עמם הארגון מתמודד ודירוג חומרת הבעיות. ה - SIM אף עוזר בניהול ומעקב אחר מהלך טיפול הטיפול בתקריות אבטחת המידע (Incident Management). כוחן של מערכות אלה טמון ביכולתן לבצע הצלבות מידע (קורלציה) שמקורו ממערכות שונות. בעזרת הקורלציה ניתן לזהות קשר בין האירועים השונים שדווחו בזמנים שונים וממערכות שונות ולהפוך אותן אקראי של אירועים (events) לתקרית (Incident) בעלת משמעות עסקית לארגון.

תקנות שונות מזכירות במפורש את הצורך בהקמת מערך לניהול אירועי אבטחת מידע, כאשר הרלוונטית מכולם לפרויקט היא הוראה 357 של המפקח על הבנקים [(9) הוראה 357 של בנק ישראל].עיקרי ההוראה מופיעים בנספח ו' - ניהול בנקאי תקין, הוראת בנק ישראל 357 - ניהול טכנולוגיות מידע.

מאיזה כיוון שמים את ה - SIM ?

הגישה שהייתה מקובלת בפרויקטים שיצאו לדרך עם תחילת התפתחותו של עולם ה - SIM התבססה על תפיסת Bottom - Up, הגורסת כי ראשית יש להתקין את המערכת על כל התשתיות בארגון, ואז ... נראה. גישה זו דורשת עבודה רבה בפריסה, התקנה והפעלה של טכנולוגיות שונות ברחבי הארגון. בסיום עבודה זו עלול הארגון למצוא את עצמו עם מערך טכנולוגי מפואר, אך לא בהכרח עם מערכת שעונה על היעדים אשר הוצבו לפרויקט.

כדי להבטיח שפרויקטי SIM יפיקו ערך עסקי אמיתי יש צורך בגישה שונה. יש צורך לנתח את מערכות ה - SIM באופן המונחה סיכונים עסקיים, בגישת Top - Down. מתודולוגיה אשר מטרתה לסייע בניתוח של מערכות SIM ברוח זו - DSOC - פותחה על בסיס הניסיון הרב שנצבר בליווי יישום מערכות SIM בישראל. במקום למהר ולפרוש את המערכת לרוחב הארגון במיקוד טכנולוגי, יש לתכנן ולבצע פרויקטים אלה על בסיס הסיכונים העסקיים ורמת הקריטיות שלהם.

התהליך מתחיל בניתוח המערכות הקריטיות והסיכונים העומדים בפניהן, עבור כל אחת מן המערכות הללו יש לזהות את מקורות המידע מהם ניתן לאסוף חיוויים רלוונטיים למצב אבטחות המידע של המערכת. מקורות המידע מורכבים ממספר שכבות: מערכות התקשורת ואבטחת המידע, בסיס הנתונים ומערכות ההפעלה וכמובן – האפליקציות עצמן.

איזה חוקים צריך ה - SIM ?

המושג "חוקים" נמצא בשימוש רב בהקשר למערכות SIM. ספקים נוהגים להעיד על כמות החוקים הקיימת בטכנולוגיה שלהם "מן הקופסא" כחלק מיתרונות המערכת. חלק מן האיומים בפניהם ניצב כל ארגון נובעים מן הטכנולוגיה המותקנת בארגון והיא משותפת לארגונים שונים. חשוב ללמוד מניסיונם של אחרים, להוציא את החוקים מן הקופסא ולהשתמש בהם לטובת הארגון. אך בנוסף לחוקים אלה, יש להגדיר במערכת ה- SIM חוקים העוזרים לזיהוי סיכונים במערכות העסקיות של הארגון אשר נובעים מניתוח הסיכונים הייחודיים הניצבים מולו [(8) גלעד, 2006].

SOC(Security operation Center)

ה - SIM מהווה חלק מתמונה רחבה יותר - מוקד אבטחת מידע (SOC - Security operation Center). מוקד זה מהווה ישות מרכזית לניהול ובקרה של אבטחת המידע של הארגון. ה - SOC מורכב מ 4 היבטים: טכנולוגיות (מערכת ה - SIM שהוצגה לעיל היא אחת מן הטכנולוגיות הנמצאות בשימוש במוקד), היבטים פיזיים (מיקום המוקד, הגנתו, ריהוטו וכדומה), האנשים העובדים בו (חשוב להקצות כוח אדם מקצועי ומנוסה לתפעול המוקד ולבנות תוכנית הדרכה מסודרת לקליטתו) ואחרון חביב - הנהלים. יש לאפיין את תהליך העבודה של המוקד באמצעות הגדרת נהלי תגובה לתרחישים המזוהים במערכת ה - SIM. נהלים אלה מגדירים את תהליכי התחקור והאסקלציה שיש לבצע בהתאם לתוכן התרחיש.

SOC ו SIM כדרך חיים

חשוב לבצע תהליך טיוב מתמיד של המוקד. אין טוב ממבחן המציאות על מנת להעריך את איכות ביצועי המוקד. יש להגדיר את האופן בו תבחן באורח שוטף איכותו של התהליך, וזאת על מנת לטייבו בהתאם ללקחים הנלמדים מן השטח. שינויים ארגוניים, הוספת מערכות חדשות, שינויים טכנולוגיים, שינויים סביבתיים, לכל אלה השפעה ישירה על הסיכונים המאיימים על מערכות המידע וכן על האיומים העומדים בפניהן. יש לעדכן את ניתוח המערכת באורח שוטף בהתאם לשינויים אלו.

שימוש במתודולוגיה בדוקה ומנוסה כמו DSOC וסיוע של צוות מקצועי המתמחה ביישום הלכה למעשה של מתודולוגיה זו ,יבטיחו של- SIM ול- SOC תהיה תרומה ישירה ליציבותו ועבודתו השוטפת של הארגון [(8) גלעד, 2006].

g. תיאור מצב קיים כולל ניתוח חלופות מערכתי

בנק דיסקונט מנהל את המידע בצורה ממוחשבת באמצעות מערכות מתקדמות שמטפלות בנתוני הלקוחות של הבנק.

מחלקת אבטחת מידע עובדת עם תוכנות מתקדמות שמאפשרות ניטור מדיניות אבטחת מידע של הבנק, כאשר קיימות מספר מערכות שרצות במקביל ובכל מערכת קיימת תוכנה שמנטרת אירועים רלוונטיים המוגדרים בפרמטרים שלה.

הגוף שאחראי על ריכוז האירועים הוא מוקד אבטחת מידע שמאויש ע"י בקרי אבטחת מידע 24 שעות ביממה, 7 ימים בשבוע. חלק מהעבודה השוטפת של בקרי המוקד היא לזהות אירועים חריגים במערכות אבטחת המידע של הבנק ולדווח על כך לגורמים הנדרשים לפי נהלי העבודה של מחלקת אבטחת מידע.

כיום בעבודה השוטפת של הבנק בכלל ומחלקת אבטחת מידע בפרט קיימים עשרות סוגים של אירועים חריגים, כאשר לכל אירוע יש להכין דו"ח מיוחד ולהעבירו לגורמים הרלוונטיים בבנק.

בעיית מצב קיים, נשארו כפי שהוגדרו בפרק 3.2 של נספח ד' - הצעת פרויקט מורחבת והן:

1. הפקת דו"חות

הפקת הדו"חות של האירועים והעברתם לגורמים הרלוונטיים בבנק לא נעשית בצורה יעילה ומהירה, זאת עקב העובדה שלוקח הרבה זמן לערוך כל דו"ח על סעיפיו השונים, כלומר קיים צורך למלא מחדש את כל שדות הדו"ח הנדרשים על מנת לשמור על הצגת הדו"ח בצורה פורמאלית שתואמת את נהלי הארגון.

2. חפיפה בין משמרות המוקד

בעת חילופי משמרות בין בקרי המוקד, החפיפה על האירועים שהתרחשו במהלך המשמרת נעשית באופן ידני, דבר זה גורם לפעמים להעברת פרטים לא מדויקים ובדרך כלל החפיפות נמשכות זמן רב.

3. חוסר בקרה

אין מערכת שתרכז את כל מגוון האירועים החריגים ביחד ותגדיר את סטאטוס הטיפול בכל אירוע חריג. דבר זה עלול להוביל למצב של התעלמות מאירועים חריגים שעדין בטיפול וטרם הסתיימו.

4. ניהול ידע

אין מערכת מרוכזת לניהול ידע שמגדירה את כל נהלי העבודה השוטפת עבור בקרי המוקד. דבר זה גורם לחוסר מקצועיות שעלול להוביל לטעויות קריטיות מצד הבקרים בזמן הטיפול באירועי אבטחת מידע חריגים.

ניתוח חלופות מערכתי

בחלק זה נציג מספר פתרונות אפשריים לפיתוח הפרויקט שפותחו ע"י חברות שונות בעולם:

1. חברת SECOZ - מערכת לניהול אירועי אבטחת מידע

חברה שמפתחת מערכות SIM - Security Information Management שמספקות לארגון תמונת מצב מקיפה אודות סטאטוס אבטחת המידע והן מאפשרות לאתר ולזהות אירועי אבטחת מידע בעת התרחשותם וכן לתחקר אותם בדיעבד. ה - SIM הינו כלי חיוני ליישום ממשל אבטחת המידע (information security governance) בארגון: לא ניתן לנהל סיכונים ללא יכולת לזהות אותם בזמן אמת ולנתח אותם לאחר מעשה.

SECOZ עוסקת בנושא זה מיום הקמתה והיא מעורבת בתכנון, ניתוח, יישום ותיעוד של מעל 15 מערכות SIM בישראל ויש לה הכרות מעמיקה עם כל הטכנולוגיות הקימות בישראל בתחום זה.

יתרונותיה של חברת SECOZ באים לידי ביטוי בהענקת שירותים לניהול אירועי אבטחת המידע הכוללים:

- כתיבת בקשה להצעות מחיר לפרויקט - פירוט הדרישות הטכנולוגיות, העסקיות והפרויקטאליות מן הספק.
- ניתוח מעני ספקים והגשת המלצות - קריאה, ניתוח טכני ועסקי של המענים, פגישות עם הספקים להצגה מקצועית והבהרות של המענים, הכנת מסמך פנימי לבדיקה להערכת המענים, הצגת הניתוח להנהלת הארגון.
- ניתוח מקורות מידע תשתיתיים - אפיון ומיפוי כל מקורות המידע. יצירת טבלה המכילה רשימה מפורטת של מקורות אלה (יכולה לשמש כקלט גם למיפוי נכסי המידע).
- ארכיטקטורה כללית של הפיתרון - מיפוי הקשרים בין מקורות המידע, שרטוט פריסת הרשת תוך התייחסות לאיסוף התראות אבטחת המידע מן המערכות השונות. ניתוח אפיון החוקים אשר יישומו על כל אחד מסוגי מקורות המידע. הצגת החוקים בצורה פורמאלית בטבלאות ניתוח הכוללות חוקי אוטנטיקציה, אוטוריזציה, ניהול מערכת, ניהול משתמשים ואבטחת מידע.
- פיתוח החוקים העסקיים - מערכת SIM אינה תורמת דבר לארגון אם לא נותחו אירועי אבטחת המידע כך שאוסף גדול של התרעות בדידות יהפכו לאירוע (incident) בעל משמעות עסקית לארגון. על מנת לאפיין את מערכת ה - SIM אנו מבצעים תהליך ניתוח המתבצע בשני צירים: ניתוח עסקי המבוסס על הסיכונים העסקים העומדים בפני הארגון והמערכות שבו וניתוח סיכונים טכנולוגים. בסיום שלב זה אנו מספקים תיאור פורמאלי של חוקי הקורלציה תוך שימוש בכלים ייחודיים שפתחנו למטרה זו.
- ליווי ישום חוקים - עבודה צמודה עם המיישם בכדי לוודא יישום התואם את צרכי א', התאמה והרחבה של החוקים לפי הצרכים בשטח.
- כתיבת תיעוד למערכת ה - SIM - הכוללת תיאור טכני ופונקציונאלי וכן נהלי תפעול וטיפול באירועים.
- שיפור מתמיד - במהלך הפרויקט אנו מבצעים ניתוח של המערכות האפליקטיביות ועידון של חוקי הקורלציה בשאר המערכות ומגדירים תסריטי תגובה, ע"מ שהמערכת תוכל לספק למנהל אבטחת המידע תמונת מצב ברורה אודות מצב אבטחת המידע בארגון.

2. חברת securitree - ניהול ותחזוקה של מערכת אבטחת מידע

החברה מתמחה בתחזוקת מערכות אבטחת מידע שמתאפיינת בהתמודדות יום יומית עם כמות אדירה של אירועים הדורשים סינון ותגובה. בנוסף יש לעדכן ולהתעדכן ברשימת עדכוני התוכנה ולעדכן את המערכות בהתאמה. נתונים אלו מביאים את ההתייחסות לתחזוקת המערכות וניהולם בתור גורם חשוב ולעיתים מכריע בקבלת ההחלטות לפתרון זה או אחר. בבחינת הפתרונות אשר החברה מציעה לארגון היא בוחנת את מידת שילוב הפתרון המוצע בתשתית הקיימת והן את יכולת ההתממשקות למוצרי ניהול מרכזיים. ראיה נוספת היא הצעה של פתרונות מרכזיים לניהול של כלל מערך אבטחת המידע או חלקים ממנו. הפתרון יכול להיות ע"י בנית מערכת פנימית או ע"י פתרון של גורם חיצוני (Outsourcing).

מגמה שהולכת ומתחזקת כיום היא גיבוש מוצרי Gateway למערכת אחת רצוי ב Appliance, וגיבוש ניהול שאר המוצרים למערכת ניהול מרכזית שתדע לנטר, לבצע correlation, להתריע ולנהל את כלל אירועי אבטחת המידע בארגון. חברת Securitree מציעה מגוון פתרונות אלו לארגונים השונים ומתאימה אותם לארגון ע"פ צרכיו ומורכבותו.

שירותי ניהול מרוחקים

חברת סקיוריטרי מציעה שירותי ניטור וניהול (Managed Security Services - MSS) במרכז האבטחה (Security Operation Center - SOC) שהקימה בישראל תוך שיתוף פעולה עם חברת Symantec , שעל בסיס הטכנולוגיות שלה מושתתות מערכות הניטור. כוח האדם המקצועי והמנוסה של שתי החברות מספק שירותי ניטור וניתוח רצופים, 24x7, אודות אירועי אבטחה המתרחשים בזמן אמת במערכותיו של הלקוח. התראות המועברות ללקוח משקפות אירועים המעידים על פוטנציאל לסיכון.

יתרונותיה של חברת סקיוריטרי באים לידי ביטוי בשירותי הניטור והניהול המסייעים ללקוח בהיבטים הבאים:

- אפיון שירותי ניטור וניהול המבוססים באופן ייחודי על צורכי הלקוח, בהתאם לפרמטרים הארגוניים.
- הגדרה נאותה של מדיניות ומוצרי האבטחה של הארגון בעזרת שירותי ניהול מקצועיים.
- התגוננות בזמן אמת מפני התראות אודות אירועי אבטחה המתרחשים בארגון.
- צמצום עלות ההשקעה במשאבי אנוש וכסף בכל הנוגע לאבטחת המידע של החברה תוך ניתוב הטיפול באירוע אבטחה לידי כוח אדם מיומן ומקצועי.
- העלאת רמת האבטחה תוך צמצום פוטנציאל הסיכון לפגיעה במידע הקיים במערכות המחשוב של ארגון הלקוח.
- מתן יכולות צפייה של הלקוח בזמן אמת, בסטאטוס האבטחה של הארגון וקבלת כלים המצביעים על הצורך בנקודות לשיפור ובמגמת השינוי במצב האבטחה בארגון.

3. חברת CA – eTrust Audit לניהול אירועי אבטחת מידע

מוצר החברה מאפשר ניהול ומעקב אחר אירועים המתרחשים בעולם המחשוב המבוזר הכולל מחשבים ואפליקציות רבות ושונות הינם אתגרים מורכבים ביותר. קיימים מנגנונים רבים להתראה על אירועים המתרחשים במערכת: מנגנוני מעקב האירועים המסופקים עם מערכות ההפעלה הינם חלקיים ומוגבלים ומספקים במקרים רבות הודעות בלתי מובנות ובלתי שמישות. כל מחשב NT או UNIX אוגר נתונים על האירועים במערכת בקבצי LOG. קבצים אלה הם מקומיים על כל מחשב והם נוטים לגדול ולנפח את שטח הדיסק המקומי. הפורמט של ההודעות אינו סטנדרטי ומנגנוני החיפוש וההתראה הינם מוגבלים. מכל האמור לעיל נובע כי ברוב האתרים מנגנונים אלה הינם מנוטרלים ואין למעשה מנגנון התראה על אירועים חריגים במערכת.

eTrust Audit מרכז בצורה אוטומטית אינפורמציית בקרה ממחשבי UNIX ו NT ומאחסן אותה בבסיס נתונים מרכזי. מנהלי המערכת יכולים להשתמש במידע זה על מנת לקבל התראות על אירועים חריגים וכן לבחון בזמן אמיתי את הפעילות המתרחשת בפלטפורמות השונות.

יתרונות eTrust Audit

- איסוף ממקורות מידע רבים למקום מרכזי אחד.
- מנגנוני התראה חזקים לאירועים נבחרים.
- מנגנוני סינון מתוחכמים לאיסוף וצפייה.
- ממשק משתמש גרפי לבחינת אירועים ודיווח.
- יכולת לגדול בשטח ולנהל אלפי תחנות NT ו UNIX.

4. פתרון נוסף שיכול לשמש כחלופה לפרויקט הוא פיתוח תוכנה דומה עם אותה הפונקציונאליות ע"י **מחלקת פיתוח של הבנק**.

יתרונות:

- מחלקת הפיתוח מכירה את מבנה הארגון ואת הצרכים שלו.
- מחלקת הפיתוח מכירה את הטכנולוגיות שאיתם עובד הבנק.

השוואה כמותית של המערכות לניהול אירועי אבטחת מידע:

ההשוואה תהיה בין המערכות הבאות:

1. מערכת SECOZ
2. מערכת Securitree
3. מערכת eTrust Audit של CA
4. המערכת שתפותח של הפרויקט

קריטריונים להשוואה :

1. עלות
2. התאמה לצרכי הלקוח
3. אמינות
4. ידידותיות ממשק
5. ביצועים

קטגוריה	SECOZ	Securitree	eTrust Audit	פרויקט
עלות	3	3	2	5
התאמה לצרכי הלקוח	4	4	4	5
אמינות	3	3	4	2
ידידותיות ממשק	4	4	5	5
ביצועים	4	4	4	3
ציון מסכם	3.6	3.6	4	4

טבלה 1 - השוואה כמותית של המערכות לניהול אירועי אבטחת מידע

הסבר חישובים בטבלה:

משקל כל סעיף בטבלה הוא 20% כאשר ניתן לשים ציון בכל משבצת בין 0 לבין 5. בציון המסכם משקללים את הציונים של כל חברה בנפרד ומקבלים ציון משוקלל מתוך 100%. לבסוף בוחרים את החברה עם הציון המסכם ה גבוהה ביותר והיא החלופה הטובה ביותר.

מסקנות השוואה כמותית:

החלופות הטובות ביותר הן מערכת eTrust Audit של חברת CA עקב ציונים גבוהים בקטגוריות ידידותיות הממשק, אמינות והתאמה לצרכי הלקוח ומערכת הפרויקט עקב ציונים גבוהים בעלות המוצר, התאמה לצרכי הלקוח וידידותיות הממשק.

מאחר והלקוח אינו מעוניין בהוצאה כספית, הוחלט לפתח את הפרויקט במסגרת פרויקט הגמר.

2. דרישות המערכת (Software Requirements)

דרישות המערכת כפי שניתן לראות בנספח ו' – הצעת פרויקט הן:

דרישות מידע ופונקציונאלית

- המערכת תגדיר תבנית דו"ח מיוחדת לכל סוג של אירוע אבטחת מידע חריג.
- המערכת תאפשר להגדיר סטאטוס עבור כל אירוע ואת אופן הטיפול בו.
- המערכת תכלול בסיס נתונים שיכיל את כל האירועים שקיימים במערכת.
- המערכת תכיל פורטל לניהול ידע שירכז את כל הנהלים הרלוונטיים לעבודה השוטפת.
- המערכת תאפשר שליטה ובקרה באמצעות ניהול מעקב על אופן הטיפול בכל אירוע חריג
- דרישה פונקציונאלית נוספת שהתווספה בזמן פיתוח הפרויקט היא מודול נוסף שינהל את סידור העבודה של הבקרים במוקד אבטחת המידע

דרישות שימושיות ואנושיות

- המערכת צריכה להיות ידידותית למשתמש ונוחה להפעלה.
- למערכת יהיה מצורף מדריך למשתמש.

דרישות ביצועים ודרישות מבצעיות

- במערכת ניתן יהיה לבצע סינון דו"חות לפי הפרמטרים הנדרשים.
- המערכת תמיין את האירועים בהתאם לסוג המערכת שממנה הגיעה התראת אבטחת מידע חריגה.

דרישות בטחון ואבטחה

- לכל משתמש במערכת יהיה שם משתמש וסיסמא ייחודיים.
- לכל משתמש יהיה סוג הרשאה המתאימה לתפקיד.
- חל איסור מוחלט להוציא דו"חות מערכת מחוץ לתחומי העבודה.
- בסיס הנתונים של המערכת יהיה מאובטח וגישה אליו תתאפשר בהתאם לסוג ההרשאה של משתמש המערכת.

דרישות תחזוקה ותמיכה

- אחת לחודש יש צורך לבצע גיבוי של בסיס הנתונים במערכת.

דרישה פונקציונאלית נוספת שהתווספה בזמן פיתוח הפרויקט היא מודול נוסף שינהל את סידור העבודה של הבקרים במוקד אבטחת המידע.

3. אפיון המערכת (Software Specifications)

a. מודל המערכת

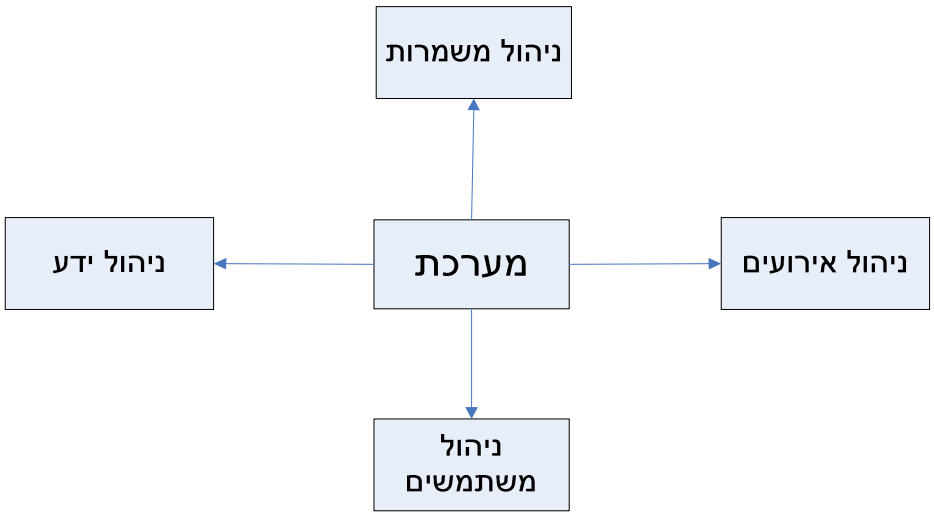
המערכת מכילה 4 מודולים מרכזיים:
1) ניהול אירועים
2) ניהול משתמשים
3) ניהול ידע
4) ניהול משמרות

חשוב לציין שחל שינוי באפיון המערכת בשלב הפיתוח לכן יש הבדל בין המודולים שהוגדרו בנספח SRD הראשוני (במסגרת דו"ח ביניים 1) למודולים הקיימים במערכת. להלן השינויים:

במסמך SRD הוגדרו 5 מודולים שהם:

1. מודל הרשאות גישה למערכת.
2. מודל תיעוד וסיווג אירועים + אופן טיפול בכל אירוע.
3. מודל חיפוש אירועים במאגר אירועים חריגים.
4. מודל הפקת דו"חות אירועים.
5. מודל לניהול פורטל ידע.

במערכת המודולים מאורגנים בצורה שונה כאשר ישנם 4 מודולים מרכזיים שהם:



איור 2 – מודולים עיקריים במערכת המידע

1. מודול לניהול אירועים שמכיל בתוכו את מודולים 2-4 שהוגדר בנספח SRD.
2. מודול לניהול ידע שמכיל את מודול 5 שהוגדר בנספח SRD.
3. מודול ניהול משתמשים שהוא מודל חדש שהתווסף בזמן פיתוח ולא הוגדר במסמך זה. הוא מכיל בין היתר את מודול 1 שהוגדר בנספח SRD.
4. ניהול משמרות שזהו מודול חדש שהתווסף תוך כדי פיתוח המערכת.

פירוט מלא של כל המודולים הסופיים, כמו גם תרשימים נלווים, ניתן לראות בנספח א' של הדו"ח - מסמך SRD.

b. אפיון פונקציונאלי

אפיון פונקציונאלי נשאר כפי שהוגדר בפרק 3.1 בנספח א' – מסמך SRD

- רישום אירוע חריג במערכת.
- הגדרת סטאטוס עבור כל אירוע והצגת אופן הטיפול בו.
- הגדרת סטאטוס אירועים חריגים בזמן אמת.
- הגדרת אופן טיפול באירוע שנרשם במערכת.
- חיפוש אירוע לפי פרמטר ספציפי.
- סינון והפקת דו"חות אירועים חריגים.
- הפקת דו"חות חפיפה בין משמרות הבקרים במוקד.
- ביצוע מעקב אחרי דו"חות החפיפה של המערכת.
- קליטת עובד חדש במוקד ועדכונו כמשתמש מערכת.
- ניהול פורטל ידע שירכז את כל הנהלים הרלוונטיים לעבודה השוטפת.
- עדכון מסמכי נהלים בפורטל הידע.
- מתן שליטה ובקרה באמצעות מעקב מסודר של טיפול באירוע חריג

c. ביצועים עיקריים

אפיון פונקציונאלי נשאר כפי שהוגדר בפרק 3.2 בנספח א' – מסמך SRD

המערכת תרוץ בשלב הראשוני במחשב אחד עם בסיס נתונים מקומי כאשר זמן התגובה של הפעולות יהיה לפי הזמנים הבאים:

עדכון פעולת רישום אירוע: עד 5 שניות.
שאילתות מערכת: עד 10 שניות בהתאם לגודל השאילתא.
העלאת מסך טיפול באירוע: עד 10 שניות.
הפקת דו"חות: עד דקה בהתאם לגודל הדו"ח.

המקום הדרוש לבסיס הנתונים יגדל עם הזמן ולא צפויים עומסים מיוחדים על המערכת.

d. אילוצי סביבה

אילוצי הסביבה נשארו כפי שהוגדרו בפרק 3.2 בנספח א' – מסמך SRD

המערכת מיועדת לעבודה השוטפת של מוקד אבטחת מידע הוא חלק ממחלקת אבטחת המידע שהיא אחת ממחלקות המטה של בנק דיסקונט. ניתן לראות את מבנה הארגון בצורה מפורטת בפרק 2 של נספח ד - הצעת פרויקט מורחבת.

e. ניתוח חלופות טכנולוגיות

בפרק זה נערוך השוואה בין החלופות הטכנולוגיות הבאות:

קטגוריה	חלופה 1	חלופה 2	חלופה 3
שפת תכנות	C#.NET	VB.NET	JAVA
סביבת עבודה	Microsoft Visual Studio .NET	Eclipse	Intellij IDEA
בסיס נתונים	SQL Server	Access	Oracle

טבלה 2 - חלופות טכנולוגיות אפשריות

השוואה כמותית של חלופות טכנולוגיות:

השוואה בין שפות תכנות:

1. C#.NET
2. VB.NET
3. JAVA

קריטריונים להשוואה :

1. בניית GUI
2. עלות
3. ידידותיות ממשק
4. זמינות חומר לימודי
5. ביצועים

קטגוריה	C#.NET	VB.NET	JAVA
בניית GUI	5	4	4
עלות	3	3	5
ידידותיות ממשק	4	4	3
זמינות חומר לימודי	5	4	4
ביצועים	5	4	5
ציון מסכם	4.4	3.8	4.2

טבלה 3 - השוואה כמותית בין שפות תכנות

הסבר חישובים בטבלה:

משקל כל סעיף בטבלה הוא 20% כאשר ניתן לשים ציון בכל משבצת בין 0 לבין 5. בציון המסכם משקללים את הציונים של כל חברה בנפרד ומקבלים ציון משוקלל מתוך 100%. לבסוף בוחרים את החברה עם הציון המסכם ה גבוהה ביותר והיא החלופה הטובה ביותר.

מסקנות השוואה כמותית:

קטגוריה	חלופות אפשריות	חלופה מועדפת	הסבר לעדיפות
שפת תכנות	1. C#.NET 2. VB.NET 3. JAVA	C#.NET	החלופות הטכנולוגית הטובה ביותר היא C#.NET עקב ציונים גבוהים בקטגוריות ביצועים, בניית GUI וזמינות חומר לימודי. שפת C#.NET היא שפה הכי נוחה לפיתוח של מערכת מידע באמצעות GUI. מבחינת ביצועים זוהי השפה נוחה מאוד לפיתוח חלקים לוגיים וגרפיים של הפרויקט.בנוסף חשוב לציין שזוהי השפה שבה משתמשים במחלקת פיתוח של הבנק לצורך פיתוח אפליקציות נוספות הקיימות בבנק. שימוש בשפה זו יאפשר בעתיד אפשרות התממשקות נוחה וקלה יותר של מערכת המידע למערכות אחרות הקיימות בבנק. שפות VB.NET ו JAVA פחות יתאימו במידה ונרצה להתממשק בעתיד למערכות אחרות בבנק.

טבלה 4 – מסקנות השוואה כמותית בין שפות תכנות

השוואה בין סביבות פיתוח:

1. Microsoft Visual Studio .NET
2. Eclipse
3. Intellij IDEA

קריטריונים להשוואה :

1. נוחות התקנה
2. עלות
3. ידידותיות ממשק
4. אמינות
5. ביצועים

קטגוריה	Microsoft Visual Studio .NET	Eclipse	Intellij IDEA
נוחות התקנה	5	4	4
עלות	3	5	3
ידידותיות ממשק	5	3	4
אמינות	5	3	4
ביצועים	5	4	5
ציון מסכם	4.6	3.8	4

טבלה 5 - השוואה כמותית בין סביבות פיתוח

<u>הסבר חישובים בטבלה:</u>

משקל כל סעיף בטבלה הוא 20% כאשר ניתן לשים ציון בכל משבצת בין 0 לבין 5. בציון המסכם משקללים את הציונים של כל חברה בנפרד ומקבלים ציון משוקלל מתוך 100%. לבסוף בוחרים את החברה עם הציון המסכם ה גבוהה ביותר והיא החלופה הטובה ביותר.

<u>מסקנות השוואה כמותית:</u>

קטגוריה	חלופות אפשריות	חלופה מועדפת	הסבר לעדיפות
סביבת פיתוח	1. Microsoft Visual Studio .NET 2. Eclipse 3. Intellij IDEA	Microsoft Visual Studio® .NET	החלופות הטכנולוגית הטובה ביותר היא Microsoft Visual Studio .NET עקב ציונים גבוהים בקטגוריות ביצועים,אמינות, נוחות התקנה וידידותיות ממשק. בנוסף נעדיף להשתמש בעורך Microsoft Visual Studio.NET כי זהו העורך המתאים ביותר לפיתוח בשפת תכנות C#.NET. שני העורכים האחרים הם עורכי JAVA בעלי פונקציונאליות זהה, ובמידה ונרצה להשתמש בשפה זו, נעדיף את עורך Eclipse עקב העובדה שעורך זה הוא חינמי לצורך פיתוח (יתרון ברור בקטגוריית העלות) לעומת Intellij IDEA שהוא בתשלום.

טבלה 6 – מסקנות השוואה כמותית בין סביבות פיתוח

ההשוואה תהיה בין בסיסי הנתונים הבאים:

1. SQL Server
2. Oracle
3. Access

קריטריונים להשוואה :

1. התאמה לפיתוח עם NET.
2. עלות
3. ידידותיות ממשק
4. אמינות
5. ביצועים

קטגוריה	SQL Server	Oracle	Access
התאמה לפיתוח עם NET.	5	4	4
עלות	4	3	3
ידידותיות ממשק	5	5	4
אמינות	5	5	5
ביצועים	5	5	4
ציון מסכם	4.8	4.6	4

טבלה 7 - השוואה כמותית בין בסיסי הנתונים

הסבר חישובים בטבלה:

משקל כל סעיף בטבלה הוא 20% כאשר ניתן לשים ציון בכל משבצת בין 0 לבין 5. בציון המסכם משקללים את הציונים של כל חברה בנפרד ומקבלים ציון משוקלל מתוך 100%. לבסוף בוחרים את החברה עם הציון המסכם ה גבוהה ביותר והיא החלופה הטובה ביותר.

מסקנות השוואה כמותית:

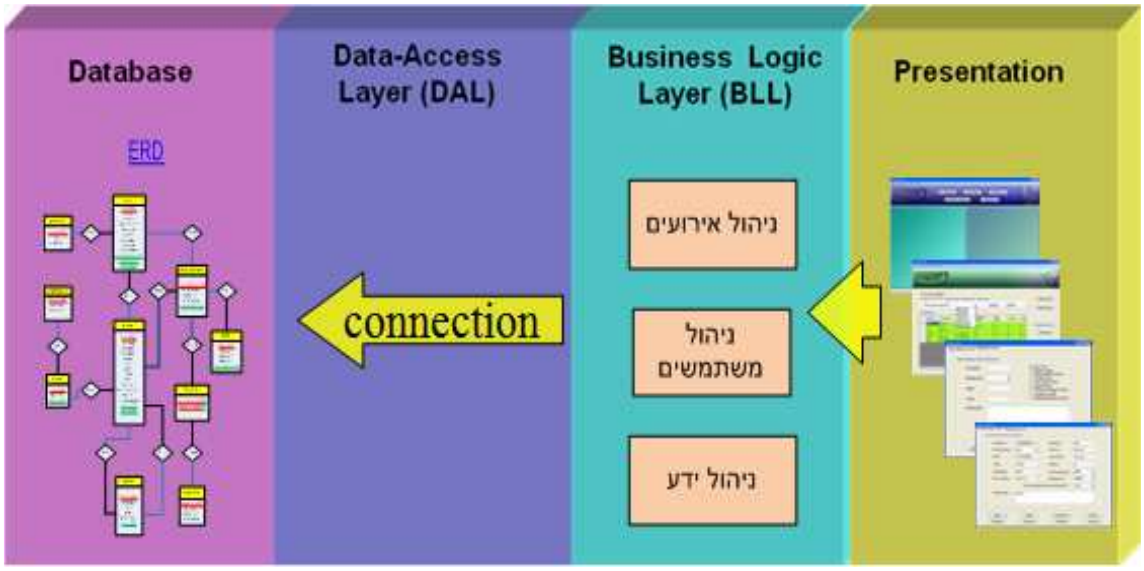
קטגוריה	חלופות אפשריות	חלופה מועדפת	הסבר לעדיפות
בסיס נתונים	1. SQL Server 2. Access 3. Oracle	SQL Server	החלופות הטכנולוגית הטובה ביותר היא SQL Server עקב ציונים גבוהים בקטגוריות ביצועים, אמינות, ידידותיות ממשק והתאמה לפיתוח עם NET. SQL Server נוח מאוד לתהליך הפיתוח וגמיש מאוד להרחבת בסיס הנתונים. בנוסף SQL Server מובנה בעורך Microsoft Visual Studio.NET, מכאן ניתן להסיק שיכולות שיתוף הפעולה בין השניים הוא פשוט ונוח. בסיסי נתונים אחרים כגון Access ו Oracle שלא מובנים בעורך, מתאימים לניהול בסיס נתונים של מערכות מידע, כאשר Oracle בהשוואה ל Access שמיועד לסביבת עבודה קטנה או בינונית, יכול לעבוד גם בסביבת עבודה גדולה. הסיבה העיקרית להעדפת SQL Server על פני Oracle הן עלות המערכת ונוחות העבודה עם עורך Microsoft Visual Studio.NET.

טבלה 8 – מסקנות השוואה כמותית בין בסיסי נתונים

4. תיכון המערכת (Software Design)

a. ארכיטקטורת המערכת

תיכון המערכת יתבצע בשיטת השכבות בצורה המתוארת באיור:



איור 3 – תיכון המערכת בשיטת השכבות

המערכת תתחלק ל 4 שכבות הבאות:

Presentation Layer – שכבה האחראית על הצגת ממשק גרפי למשתמש.

Business Logic Layer – שכבה האחראית על פעולות לוגיות של המערכת, מכילה מחלקות המייצגות את הישויות המרכזיות במערכת ואחראית על הפעלת שאילתות לצורך עבודה מול בסיס הנתונים (עבודה מול בסיס הנתונים יעשה באמצעות שאילתות דינאמיות ובאמצעות stored procedures).

Data Access Layer – שכבה האחראית על יצירת תקשורת בין שכבת הלוגיקה לשכבת בסיס הנתונים.

Database – שכבה המכילה את נתוני הטבלאות בבסיס הנתונים ופרוצדורות שמורות שמופעלות משכבת הלוגיקה.

פירוט של התהליכים בכל אחד מהשכבות ניתן לראות בפרק 2 של נספח ב' - מסמך SDD.

b. תיכון מפורט

בחלק זה נתאר את כל הקומפוננטות המרכזיות והמסכים השונים שקיימים במערכת

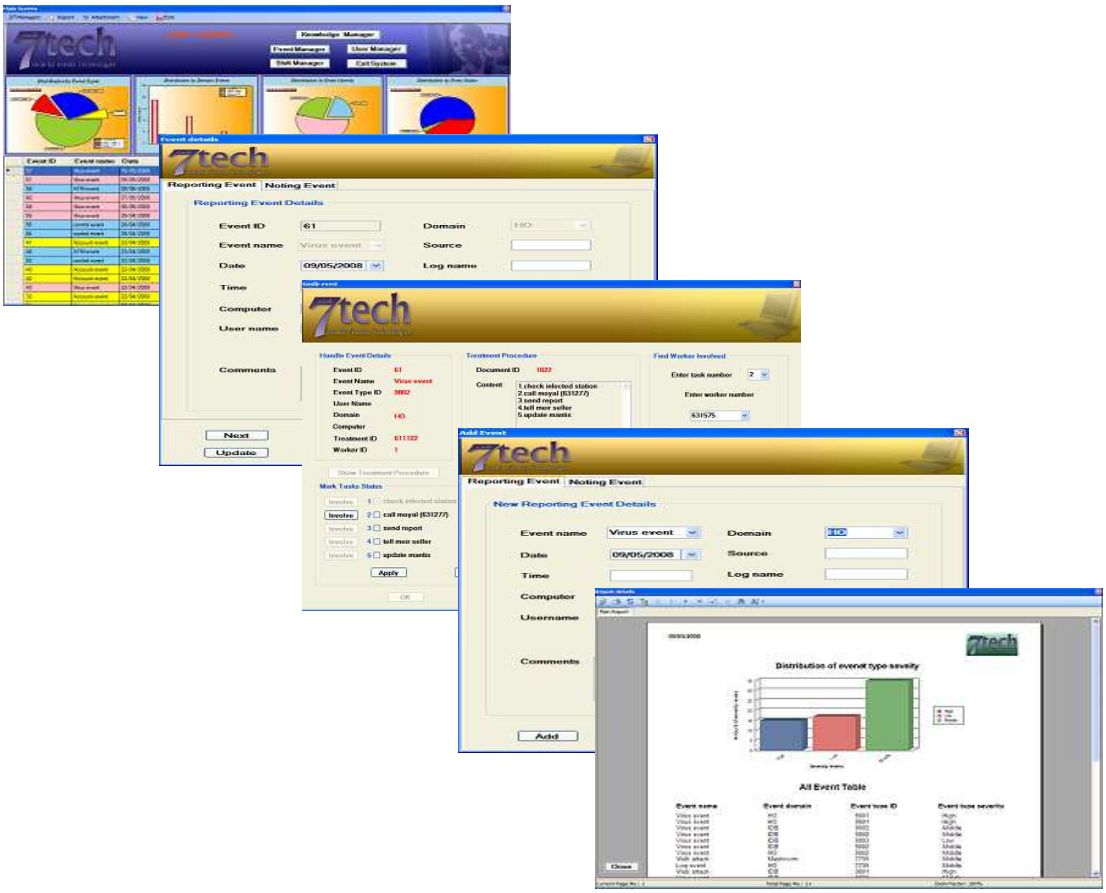
המערכת מכילה קומפוננטות המרכזיות הבאות לביצוע כל תהליכי המערכת.

- תפעול בסיס נתונים (DBManipulate)
- אירוע (Event)
- סוג אירוע (Event Type)
- משתמש (User/Worker)
- נוהל טיפול באירוע (Event Treatment Procedure)
- פריט מטופל (Treatment Item)
- שירות שכבת הלוגיקה (BLService)
- סידור עבודה (ShiftSchedule)
- אלגוריתם להתאמת נוהל טיפול (DecideTreatAlgorithm)

סקירה מפורטת של כל קומפוננט ניתן לראות בפרק 3 של נספח ב' – מסמך SDD.

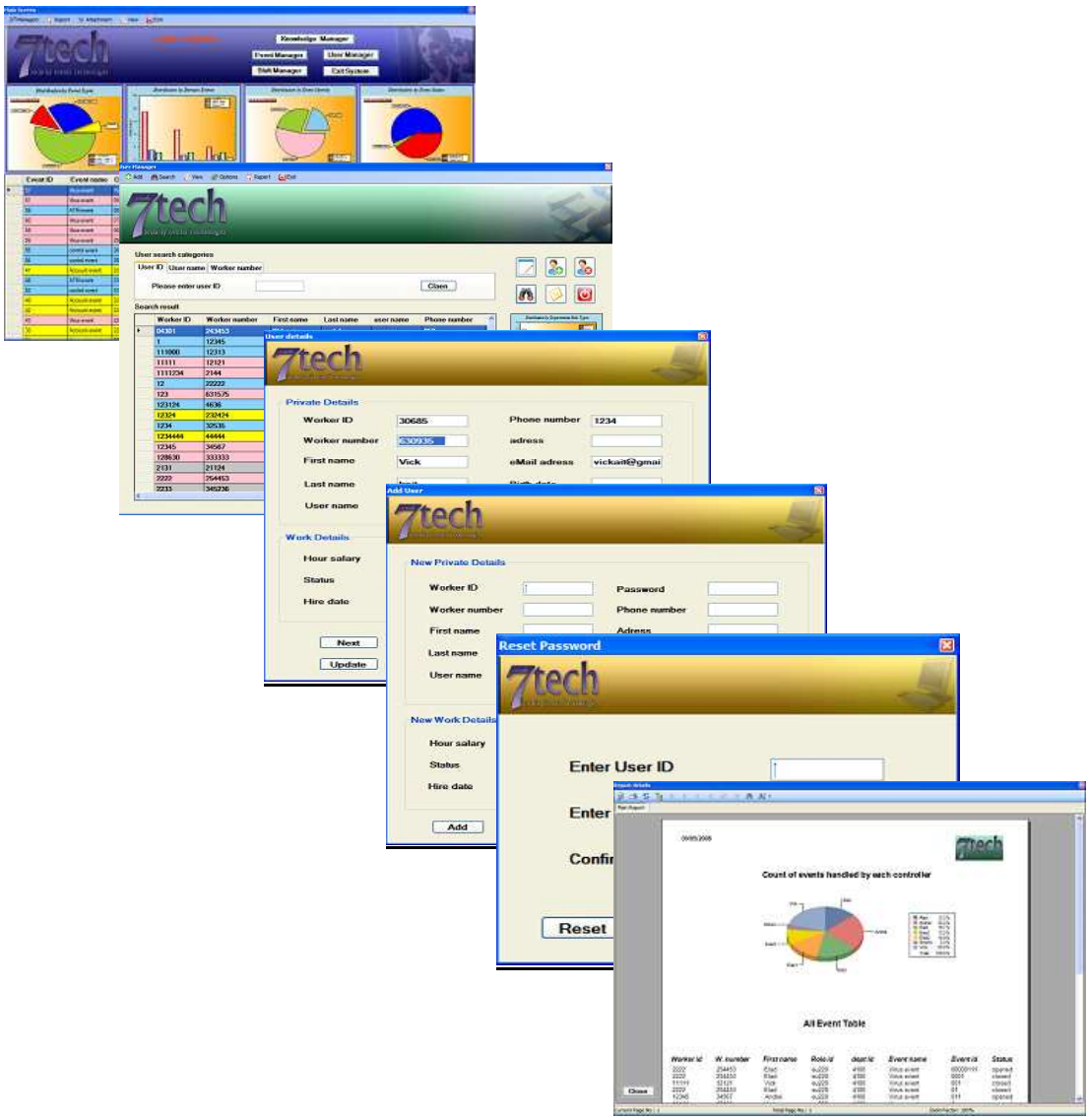
כמו כן עבור בל מודול מרכזי המערכת מכילה את המסכים הבאים:

1. מנהל אירועים



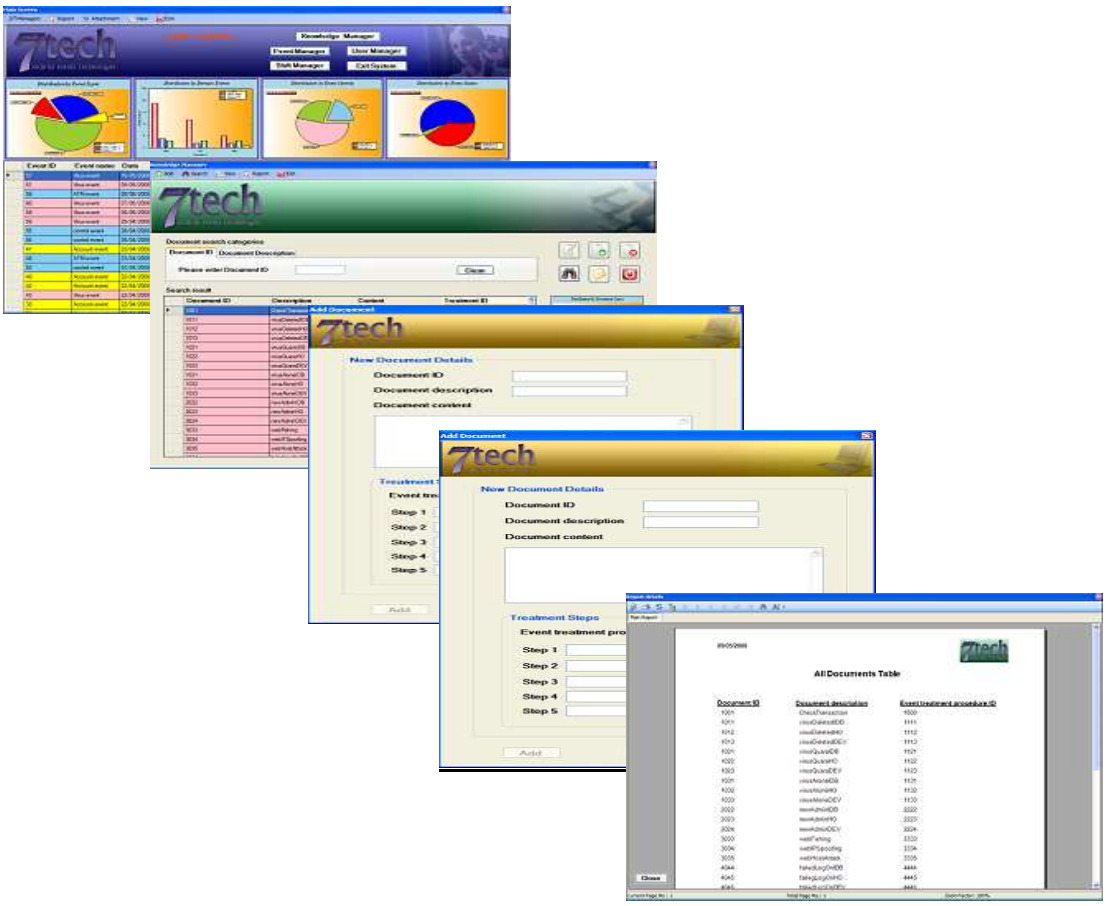
איור 4 – מסכי מודול מנהל האירועים

2. מנהל המשתמשים



איור 5 – מסכי מודול מנהל המשתמשים

3. מנהל הידע



איור 6 – מסכי מודול מנהל הידע

4. מנהל המשמרות

איור 7 – מסכי מודול מנהל המשמרות

סקירה מפורטת של כל פונקציונאליות המסכים ניתן לראות בפרק 4 של נספח ב' – מסמך SDD.

c. אלטרנטיבות לתיכון המערכת

קישור לבסיס הנתונים

האלטרנטיבות לקישור לבסיס נתונים הן: שאילתות דינאמיות או Stored Procedures.
קודם כל נגדיר כל אחד מהאלטרנטיבות:

Stored Procedures - "פרוצדורות מאוחסנות"

אלו בעצם פרוצדורות (ובמקרה שלנו - שאילתות), אשר "מתחסנות" בזיכרון השרת ע"י קימפול השאילתה לשרת, בפעם הראשונה שהיא מורצת (ובכל פעם שהטבלה מתעדכנת).

שאילתות דינאמיות

אלו שאילתות שנכתבות בשכבת הלוגיקה לצורך ביצוע מניפולציות שונות על בסיס הנתונים דרך שכבת ה DAL, שבאמצעותו מועבר קוד SQL לשכבת בסיס הנתונים.

כעת נשווה בין 2 השיטות באמצעות טבלה:

	Stored procedures	שאילתות דינאמיות
אבטחה	מונע SQL injection	לא בכל המקרים מונע SQL injection
ביצועים	יותר טובים באפליקציות גדולות (אופטימיזציה מערכתית על השאילתות, נטענת ל CACHE ומעודכנת בכל פעם שמבוצע INSERT לטבלה)	שווים ל Stored Procedures באפליקציות פשוטות אך פחות טובים באפליקציות גדולות
ארגון	קוד SQL נמצא רק בשכבת בסיס הנתונים (encapsulation בגישה לנתונים דרך component, בדרך זו נעשה גם הקישור לבסיס הנתונים)	קוד SQL נמצא בשכבת הלוגיקה ובשכבת DAL (גישה לבסיס הנתונים נעשה דרך שכבת DAL באמצעות שאילתות SQL)
תחזוקה	שינוי לא דורש recompilation של חלקים אחרים באפליקציה	שינוי דורש recompilation של חלקים אחרים באפליקציה
שינוי בסיס נתונים	משנים את כל הפרוצדורות השמורות בשכבת בסיס הנתונים עקב העובדה שכל שאילתות ה SQL נמצאת בשכבת בסיס הנתונים	משנים מספר מתודות בשכבת ה DAL שאחראית על הקישור לבסיס הנתונים, כל שאילתות ה SQL נשארות בשכבת הלוגיקה

טבלה 9 - אלטרנטיבות לקישור בסיס הנתונים

כעת נסקור את היתרונות והחסרונות של כל אלטרנטיבה:

יתרונות Stored Procedures:

1. שאילתה מאוחסנת וקבועה, ניתן לשנותה ותשפיע על כל המערכת בהתאם, ללא צורך לשנות ספציפית בכל השאילתות הזהות\מקושרות לה.
2. אפשרויות אופציונאליות רבות. כולל תנאים, הגדרת משתנים, ועוד.
3. שיתוף בין אפליקציות.
4. מימד אבטחה נוסף (אפשרות הצגת-נתונים גם למשתמשים שאין להם הרשאות לצפייה בטבלאות מסוימות).
5. אופטימיזציה מערכתית על השאילתות (נטענות ל CACHE, ומעודכנות בכל פעם שמבוצע INSERT לטבלה).

חסרונות Stored Procedures:

כאשר משנים את בסיס הנתונים (למשל: מ SQL Server ל Oracle) אז יש צורך לשכתב מחדש את כל הפרוצדורות השמורות בשכבת בסיס הנתונים עקב העובדה שכל שאילתות ה SQL נמצאת בשכבת בסיס הנתונים.

יתרונות שאילתות דינאמיות:

1. במקרה של שינוי בסיס הנתונים יש צורך לשנות בסך הכול מספר מתודות בשכבת ה DAL שאחראית על הקישור לבסיס הנתונים, כל שאילתות ה SQL נשארות בשכבת הלוגיקה.
2. שיטה פשוטה מאוד שמנהלת בצורה לא פחות טובה עבודה מול בסיס הנתונים מאשר Stored procedures באפליקציה פשוטות.

חסרונות שאילתות דינאמיות:

1. לא בכל המקרים מונע SQL Injection.
2. שיטה פחות טובה מ Stored procedures באפליקציות גדולות ומתרחבות.
3. שינוי בשאילתה דורש recompilation של חלקים אחרים באפליקציה.

לסיכום:

אני בחרתי לשלב 2 שיטות יחד כאשר עבור שאילתות מורכבות אני משתמש ב stored procedures ובשאילתות פשוטות אני משתמש בשאילתות דינאמיות.

עיצוב GUI בפרויקט

בעיצוב GUI התלבטתי בין 2 דברים מרכזיים:

1. הצגת נתוני הטבלאות מתוך ה DataGrid.
2. הצגת אופציית חיפוש נתונים.

האופציות מתוארות בטלה הבאה:

	אופציה א	אופציה ב
הצגת נתוני הטבלאות מתוך ה DataGrid	הצגת נתוני הטבלאות באמצעות לחיתה כפולה על אחד המשבצות בתוך ה DataGrid	הצגת נתוני הטבלאות באמצעות סימון שורה רצויה מתוך DataGrid ולחיצת על "הצג"
הצגת אופציות חיפוש נתונים	הצגת אופציית חיפוש באמצעות פתיחת חלון חדש לצורך הכנסת פרמטרים לחיפוש	ביצוע אופציית חיפוש בחלון המרכזי באמצעות TabContol, לצורך ביצוע חיפושים שונים בהתאם לטבלאות בבסיס הנתונים

טבלה 10 - אלטרנטיבות לעיצוב GUI בפרויקט

לסיכום:

הצגת נתוני הטבלאות מתוך ה DataGrid תעשה באמצעות לחיצה כפולה על אחד המשבצות בתוך ה DataGrid עקב העובדה שזה יותר אינטואיטיבי ויותר מקצועי, דבר זה מאפשר בין היתר הגדרת אירוע וטיפול בו בהתאם לדרישות המערכת.

הצגת אופציות חיפוש נתונים תעשה בחלון המרכזי באמצעות TabContol, לצורך ביצוע חיפושים שונים בהתאם לטבלאות בבסיס הנתונים מכיוון שבצורה זו יותר נוח לעשות מעקב על הנתונים שאתה מחפש.

5.תכנון הפרויקט (Project Planning)

a. ניהול סיכונים

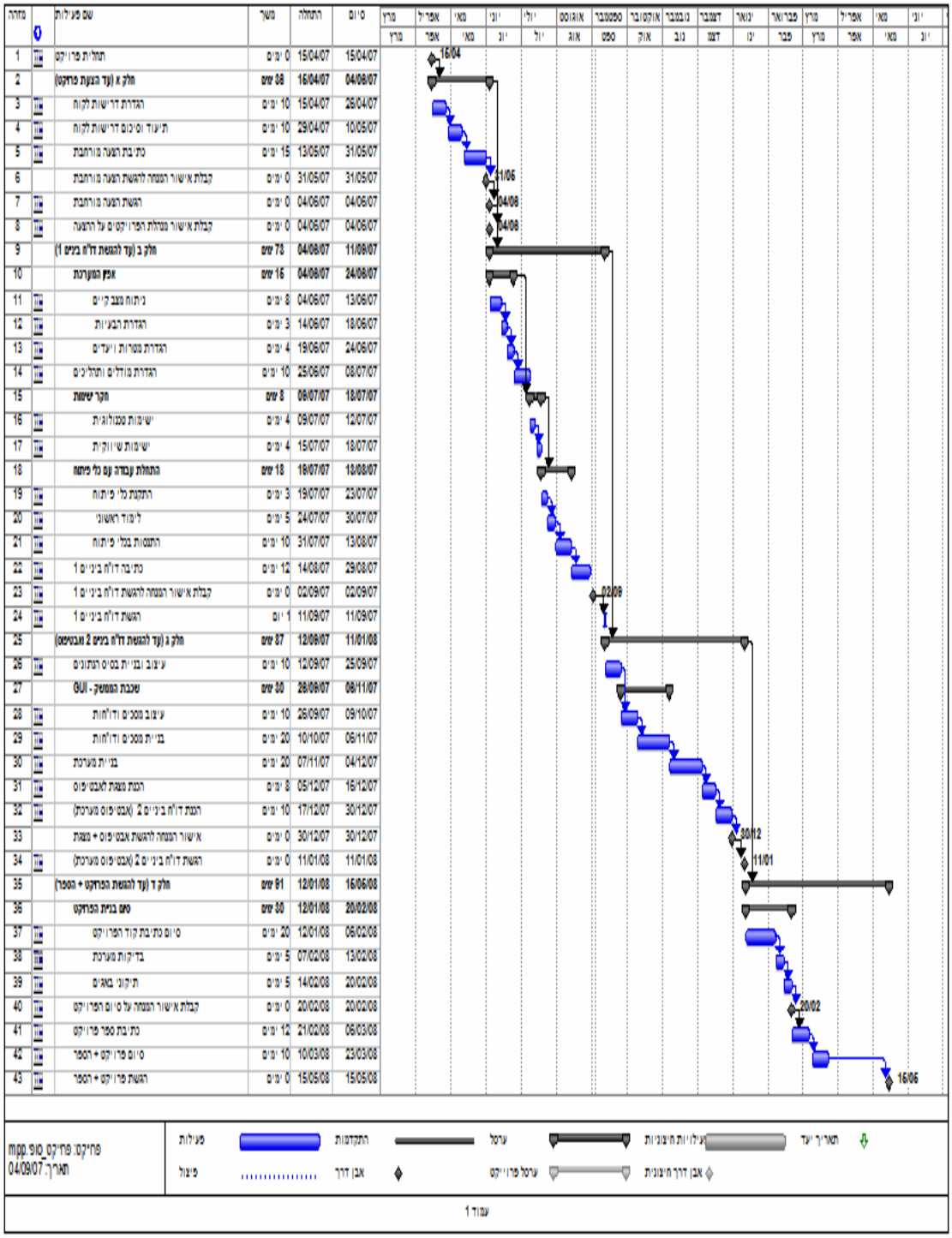
להלן ניתוח סיכונים מעודכן של הפרויקט כפי שניתן לראות בנספח ה – מסמך SPMP כאשר מאז תחילת הפרויקט לא היה שום שינוי מבחינת ניתוח והערכת סיכונים:

	קטגוריית הסיכון	תיאור הסיכון	סיכוי הסיכון	השפעת הסיכון	דרך לפתרון הסיכון
1	תכנון ראשוני בלתי מספיק	הערכה לא נכונה של גודל המערכת שיש לפתח	ממוצע (-25% 50%)	אי עמידה בלוח הזמנים של הפרויקט (רצינית)	הגדרה מפורשת של היקף הפרויקט לפני תחילתו
2	ניהול פרויקט בצורה לא נכונה	במקום התכנון והאפיון עוברים ישר למימוש	נמוך (-10% 25%)	אי שביעות רצון מהמוצר מצד הלקוח ואי עמידה בלוח הזמנים של הפרויקט (קטסטרופלית)	הגדרה ומעקב צמוד אחרי ביצוע הפרויקט כאשר יש צורך להגדיר תנאים במעבר לכל שלב כאשר חלק מאותם תנאים יהיה סיום מלא של ביצוע השלב הקודם.
3	שינויים בדרישות הלקוח	יותר שינויים בדרישות הלקוח מהצפוי	ממוצע (-25% 50%)	אי עמידה בלוח הזמנים ואי עמידה בתקציב הפרויקט (רצינית)	הגדרה מסודרת של דרישות הלקוח לפני שמתחילים בבניית הפרויקט
4	גורמים טכנולוגיים	באג מערכת בסביבת פיתוח	נמוך (-10% 25%)	אי עמידה בלוח הזמנים ואי שביעות רצון מהמוצר מצד הלקוח (קטסטרופלית)	לימוד טוב של סביבת הפיתוח כולל כל החסרונות וכאשר רואים בעיה עתידית בתהליך הפיתוח יש צורך לבדוק חלופות מתאימות
5	גורמים טכנולוגיים	מעבר לטכנולוגיה יותר מתקדמת	נמוך (-10% 25%)	אי תאימות המוצר לארגון (קטסטרופלית)	לערוך סקר טכנולוגיות לפני שמתחילים את הפרויקט ולעשות הכנה מוקדמת למקרה שבו יכולות להיכנס לשוק טכנולוגיות חדשות
6	גורמים אנושים	מחלת מפתח הפרויקט	נמוך (-10% 25%)	אי עמידה בלוח הזמנים של הפרויקט (רצינית)	אין אפשרות למנוע
7	גורמים ארגוניים שונים	שינוים במבנה הארגון	נמוך (-10% 25%)	אי תאימות המוצר לארגון או לאי עמידה בלוח הזמנים של הפרויקט (רצינית)	ללמוד טוב את מבנה הארגון עבורו מפתחים את המערכת ובדיקה מקדימה האם הארגון עלול לעבור שינויים אדמיניסטרטיביים

טבלה 11 - ניהול הסיכונים הקיימים

b. תוכנית עבודה

להלן לוח הזמנים המעודכן של הפרויקט כפי שניתן לראות בנספח ה – מסמך SPMP:



איור 8 – תרשים משימות לפי לוח זמנים בעזרת תוכנת MS-Project

6. בדיקות והערכה (Software Testing and Evaluation)

a. תוכנית בדיקות תוכנה

מסמך תכנון ועיצוב הבדיקות מוגדר לפי המודולים המרכזיים של המערכת כלומר לכל מודול מרכזי תהיה סדרת בדיקות משלו וכמובן בסופו של דבר תיערך סדרת בדיקות למערכת כולה.

הבדיקות יתבצעו על קוד המקור שבאמצעותו תפותח המערכת, כאשר הבדיקה תחולק ל 3 שלבים עיקריים:

1)בדיקות יחידה

2)בדיקות אינטגרציה

3)בדיקות מערכת

פירוט מלא של פרק תוכנית בדיקות תוכנה ניתן לראות בנספח ג' של הדו"ח – מסמך STD, מסמך שבו מתוארות תכונות המערכת שיבדקו בשלב הבדיקות, סביבה נדרשת לביצוע הבדיקות ומקרי בדיקה שונים למודולים המרכזיים של המערכת.

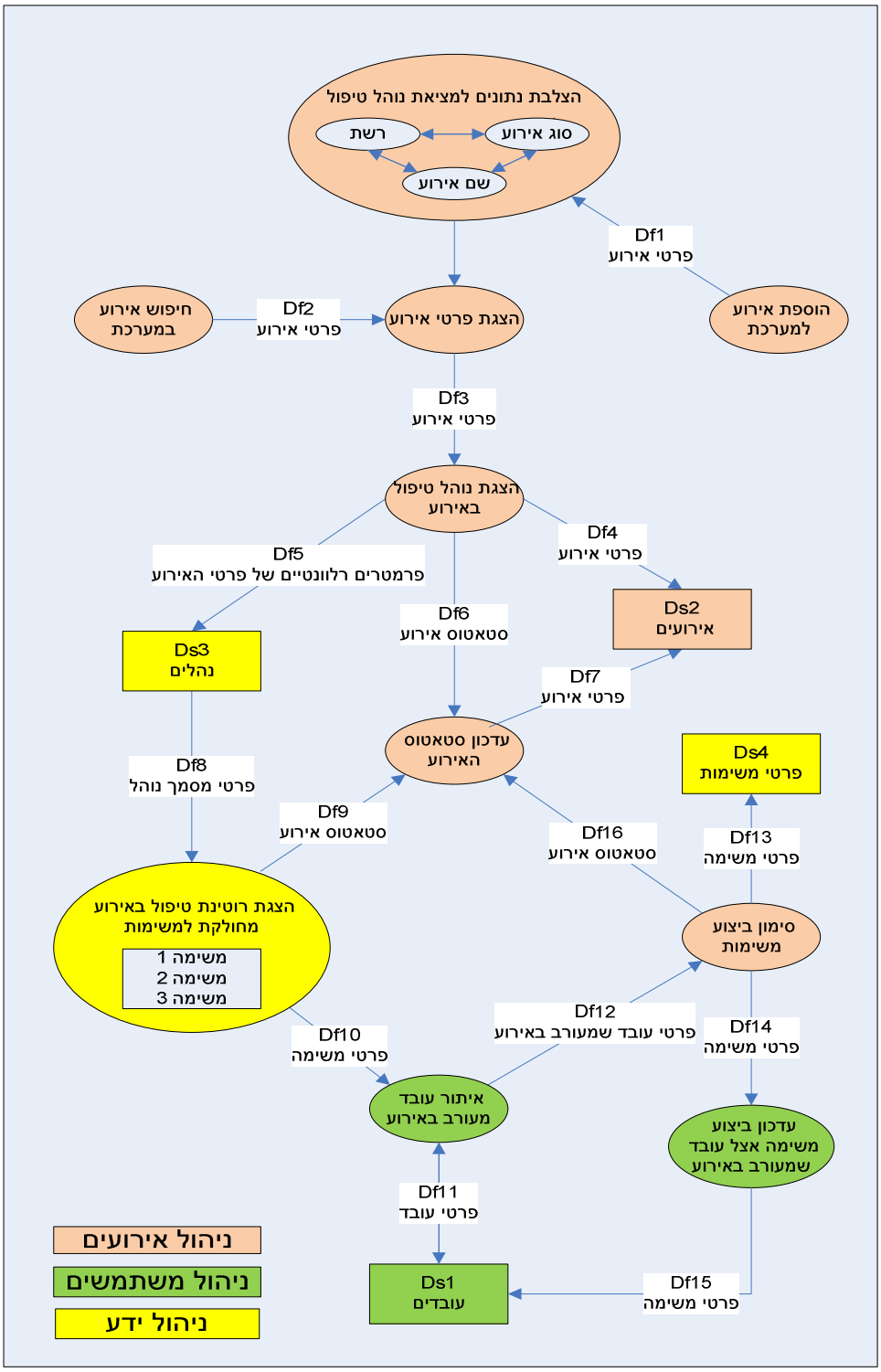
b. דוח בדיקות תוכנה

בדיקות התוכנה שהוגדרו למערכת בנספח ג' של הדו"ח – מסמך STD, התבצעו בהצלחה ואת כל התוצאות ניתן לראות בנספח ד – מסמך STR

c. דוגמאות הפעלה מפורטות מקצה לקצה

1.תהליך מעקב טיפול באירוע שנרשם במערכת

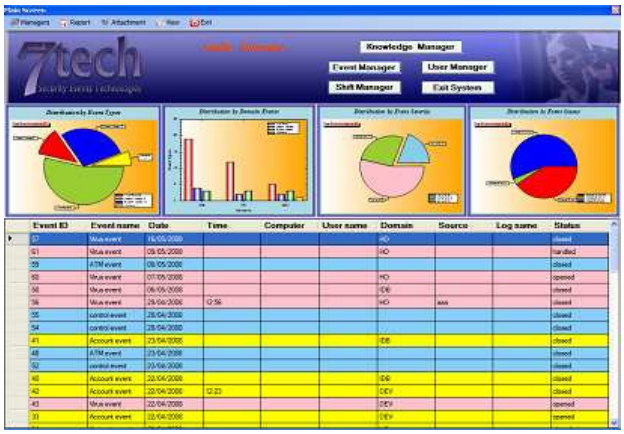
דוגמא לתהליך מרכזי להפעלת נוהל טיפול באירוע שנרשם במערכת המערב 3 מודולים עיקריים של המערכת (ניהול אירועים, ניהול ידע וניהול משתמשים)
תחילה נציג תרשים זרימה המתאר את התהליך ולאחר מכן נעבור על כל שלבי התהליך תוך כדי הצגת המסכים הרלוונטיים. להלך התרשים:



איור 9 – תרשים זרימה לתהליך מעקב טיפול באירוע שנרשם במערכת

כעת נעבור על שלבי התהליך:

1. כנס למערכת ותגיע לדף הראשי



איור 10 – מסך מערכת ראשי

2. בדף הראשי בחר את "Event Manager"

איור 11 – מסך מנהל אירועים

3. במנהל האירועים בחר "Add event" , מלא את פרטי האירוע ולחץ "Add"

איור 12 – מסך הוספת אירוע

4. כעת נוצר אירוע חדש שהסטאטוס שלו פתוח וניתן לראות אותו בתצוגת Grid במנהל האירועים של המערכת
5. בעת הוספת האירוע, המערכת מפעילה אלגוריתם שמאתר רוטינת טיפול מתאימה לאירוע על פי הפרמטרים הבאים: שם אירוע, סוג אירוע, רשת שבה אירע האירוע.
6. כעת נבחר מתוך ה Grid את האירוע החדש על מנת להפעיל את רוטינת הטיפול. לשם כך נסמן את רשומת האירוע ב Grid ונלחץ "Show Details" או פשוט נלחץ Double Click על האירוע הרצוי ונגיע למסך שמציג את פרטי האירוע

איור 13 – מסך פרטי אירוע

7. על מנת להפעיל את רוטינת הטיפול נלחץ על כפתור "Handle" ונגיע למסך הטיפול

איור 14 – מסך טיפול באירוע

8. המערכת תציג נוהל מתאים לאירוע המבוקש הכולל מסמך נוהל ורוטינה דינאמית על מנת לבצע מעקב אחר הטיפול באירוע
9. רוטינת הטיפול תחולק לתתי משימות שאותן יש לבצע לפי הסדר שהגדירה המערכת
10. במידה ויש צורך לערב עובד מסוים בטיפול, יש ללחוץ "Involve" והמערכת תאתר את פרטי העובדים שמעורבים בנוהל הטיפול באירוע ותשלח לו מייל במקרה הצורך
11. הבקר יסמן את המשימות שביצע ויעדכן את המערכת
    11.1. אם (כל המשימות בוצעו)

11.1.1. המערכת תעדכן את סטאטוס האירוע כ"סגור"
11.1.2. המערכת תעדכן את פרטי המשימה שבוצעו
11.1.3. המערכת תעדכן את סטאטוס רוטינת הטיפול
11.1.4. המערכת תתייק עובדים רלוונטיים שהיו מעורבים בטיפול

11.2. אחרת

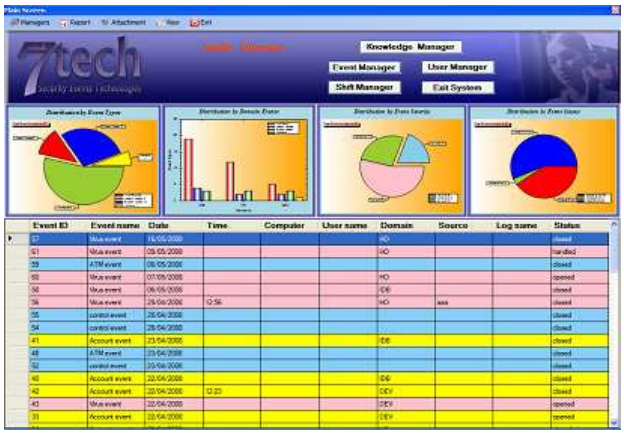
11.2.1. המערכת תעדכן את סטאטוס האירוע כ"מטופל" וניתן יהיה לחזור בשלב יותר מאוחר לרוטינת הטיפול על מנת להשלימה
11.2.2. המערכת תעדכן את פרטי המשימה שבוצעו
11.2.3. המערכת תתייק עובדים רלוונטיים שהיו מעורבים בטיפול

12. לאחר סגירת האירוע המערכת תחזור למסך מנהל האירועים .

2. תהליך הוספת רוטינת טיפול חדשה והתאמתה לפרמטרי האירוע

תהליך שמאפשר הגדרת רוטינת טיפול חדשה, שיוך מסמך נוהל לרוטינת טיפול ולאחר מכן תיוק נוהל הטיפול לאירועים רלוונטיים המתאימים לרוטינה לפי הפרמטרים הנדרשים.

1. כנס למערכת ותגיע לדף הראשי



איור 15 – מסך מערכת ראשי

2. בדף הראשי בחר את "Knowledge Manager"

איור 16 – מסך מנהל ידע

3. במנהל האירועים בחר "Add Document" ותגיע למסך הבא

איור 17 – מסך הוספת מסמך

4. בשלב זה אין אפשרות ללחוץ על כפתור "Add" לכן קודם יש למלות מספר מזהה לרוטינה ו 5 שלבים ברוטינה החדשה וללחות "Apply"
5. בשלב הבא המערכת תעתיק את שלבי הטיפול לתוך המסמך, נגדיר שם למסך ונלחץ "Add" ליצירת הרוטינה החדשה.
6. לאחר מכן נסגור התוכנית תחזור למסך "מנהל הידע" וכעת נשאר לתייק את רוטינת הטיפול לאירועים המתאימים לפי הפרמטרים הרצויים.
7. נצא מסך "מנהל האירועים" ונגיע למסך הראשי, נלך ל Tool Strip ונבחר באופציית Attachment → Attach Treatment Procedure ונגיע למסך הבא

איור 18 – מסך תיוק רוטינת טיפול

8. במסך זה יש לבחור את כל הפרמטרים הנדרשים וביניהם: שם אירוע, סוג אירוע, רשת בה אירע האירוע, הפרוצדורה עצמה וכמובן יש צורך לתת שם לפרוצדורה החדשה.
9. לאחר שנלחץ "Attach" הפרוצדורה תיתן מענה לכל אירוע שיהיו בו כל אותם הפרמטרים שהוגדרו במסך זה ותיוקו לרוטינת הטיפול הרלוונטית.

d. ניתוח יעילות

ביצועי מערכת

המערכת היא מערכת Stand Alone, לפיכך היא לא מושפעת ע"י גורמים חיצוניים, ביצועי המערכת הם ביצועים מהירים בהתאם לכמות המידע ומספר הטרנזקציות שנטענות מבסיס הנתונים.

עלות מערכת

גודל בסיס הנתונים הוא קטן כאשר הוא יגדל עם הפעילות השוטפת של המערכת, לכן יש צורך לבצע גיבויים ותחזוקה שותפת של בסיס הנתונים.

אמינות (איכות) המערכת

המערכת מבצעת נכון את מטרותיה כאשר יש לקחת בחשבון שתמיד יש אפשרות של טעויות אנוש (למשל: בהזנת נתונים למערכת).

שלמות המערכת

המערכת מבצעת את כל המטרות שהוגדרו לה.

ייחודיות ומקוריות

המערכת פותחה בהתאם לצרכים של הלקוח וכל הייחודיות שלה שהיא מותאמת לצורכי הארגון ומיעלת את העבודה השוטפת מכל הבחינות.

אבטחת מידע

קיימים מספר מנגנוני אבטחת מידע בהתאם לפרק 3.6 בנספח א' - מסמך SRD

1) לכל משתמש במערכת יהיה שם משתמש וסיסמא ייחודיים כדי למנוע גישה של אנשים שלא מורשים לגשת למערכת.

2) לכל משתמש יהיה סוג הרשאה המתאימה לתפקיד.

3) מבחינת המערכת יש מנגנון הגנה שמונע SQL Injection, כאשר יש בקרה על הקלט הנכנס לתיבות הטקסט בכל מסכי המערכת.

4) מנגנון נוסף שנועד למנוע גניבת סיסמאות של משתמשים אחרים מתוך המערכת הוא מנגנון הצפנת סיסמאות כאשר המערכת מפעילה אלגוריתם שממיר את הסיסמא למחרוזת באמצעות מערך ביטים ובצורה זו הסיסמא נשמרת בבסיס הנתונים.

7. התוצר

המערכת מכילה קוד מקור שמתחלק ל 3 פרויקטים עיקריים שכל אחד מהם מייצג שכבה נפרדת בפרויקט + בסיס הנתונים.

1. פרויקט 1 – Presentation Layer המכילה את כל המסכים והדוחות שנמצאים במערכת.
2. פרויקט 2 – Business Logic Layer המכילה את כל המחלקות והקומפוננטות המרכזיות של המערכת.
3. פרויקט 3 – Data Access Layer המכיל פרוצדורות גישה לבסיס הנתונים.
4. בסיס נתונים המכיל את כל הנתונים והטבלאות השייכות למערכת.

ניתן לראות בפירוט את כל מסכי המערכת בפרק 3 של נספח ב' – מסמך SDD.

דוחות המערכת מתחלקים לפי המודולים השונים כאשר בסך הכל יש כ - 15 תבניות שונות של דוחות מערכת המתחלקים באופן הבא:

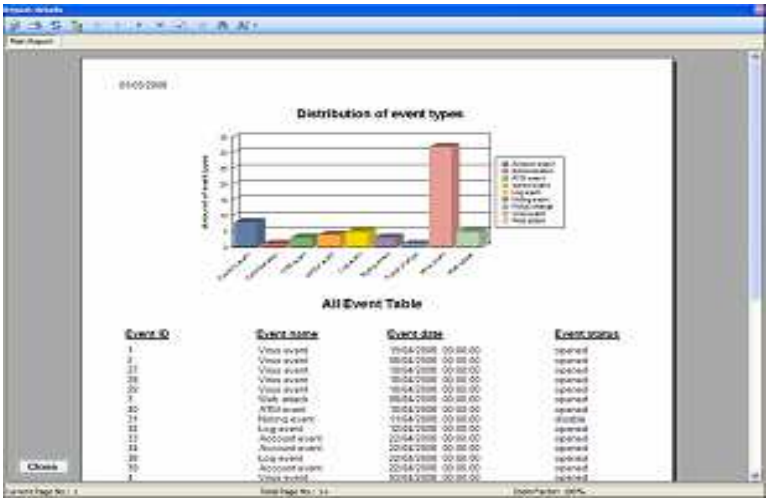
<u>מנהל אירועים</u>: דו"ח פרטי אירוע (דינמי), דו"ח אירועים לפי סוג אירוע (דינמי + רגיל), דו"ח אירועים לפי רשתות הארגון (דינמי + רגיל), דו"ח אירועים לפי חומרת האירוע (רגיל), דוח אירועים לפי סטאטוס האירוע (דינמי +רגיל), דו"ח אירועים בטווח תאריכים (דינמי), דו"ח אירועים לפי עובד שטיפל בהם (דינמי), דו"ח כל האירועים (רגיל).

<u>מנהל משתמשים</u>: דוח פרטי משתמש (דינמי), דו"ח משתמשים לפי מחלקה (דינמי + רגיל), דוח משתמשים לפי תפקיד (דינמי + רגיל), דו"ח פרטי טיפול לפי עובד שהיה מעורב בהם (דינמי), דו"ח כל המשתמשים (רגיל).

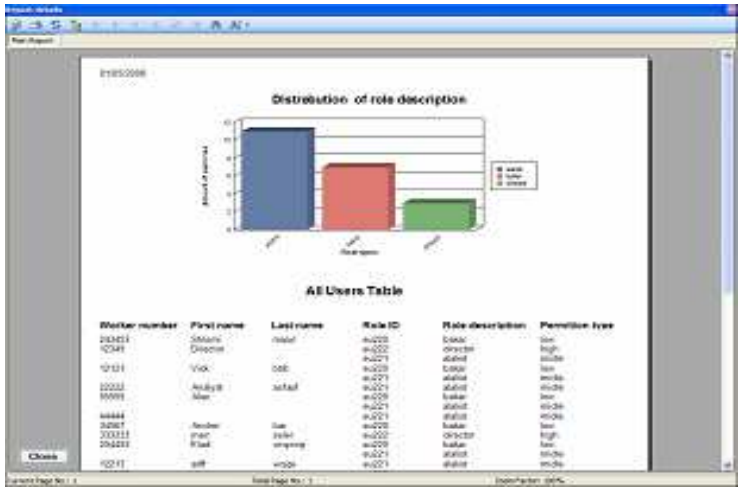
<u>מנהל הידע</u>: דו"ח כל הנהלים (רגיל), דו"ח כל פרטי הטיפול באירועים (רגיל).

<u>מנהל המשמרות</u>: דו"ח כל סידורי העבודה (רגיל), דו"ח משמרות לפי עובד (דינמי), דוח משמרות לעובד בטווח תאריכים (דינמי).

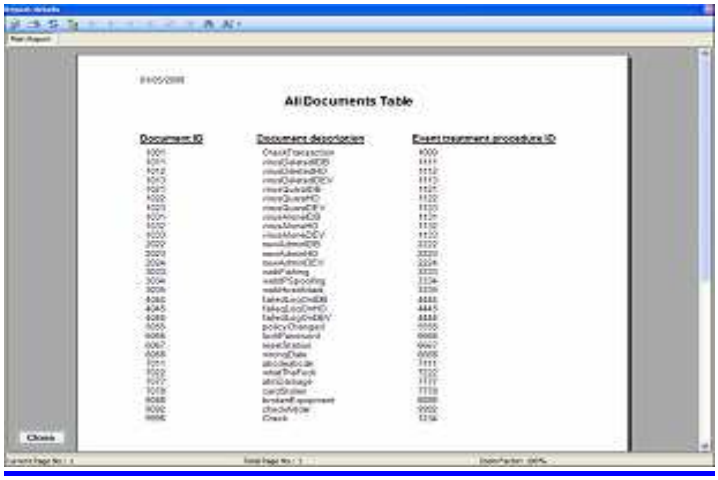
להלן מספר דוגמאות לדוחות:



איור 19 – דו"ח אירועים לפי סוג אירוע



איור 20 – דו"ח משתמשים לפי סוג תפקיד



איור 21 – דו"ח כל המסמכים

ציוד חומרה ותוכנה הנדרש להפעלת המוצר (צד לקוח)

תוכנות:

- מערכת הפעלה מסוג Windows 2000 ומעלה עם עדיפות למערכת הפעלה Windows XP Professional.
- תוכנות Microsoft Office.
- Microsoft SQL Server 2005 (עבור בסיס הנתונים).

חומרה: עבור מחשב או רשת מחשבים שמריצות את מערכת המידע:

- מעבד פנטיום 4.
- זיכרון 512MB ומעלה.
- כרטיס רשת.
- כבל רשת.

8. סיום

a. סיכום ומסקנות

תהליך העבודה היה מורכב וקשה כאשר יש צורך בלמידה עצמית של 95% לפחות מכלל הידע הנדרש לביצוע הפרויקט ולדעתי נדרשות הרבה יותר מ – 400 שעות שמקדישה המכללה לביצוע פרויקט בסדר גודל כזה, מכיוון שצריך להתחשב שהסטודנטים הם מפתחים מתחילים שצריכים תוך כדי כל שלבי הפרויקט, קודם כל ללמוד את החומר ורק אחר כל לממש וזה לוקח הרבה יותר זמן ממה שהוקצב לביצוע הפרויקט.

בעיות העיקריות בתכנון הפרויקט היו איך לחלק את המודולים המרכזיים של הפרויקט בצורה הנכונה ביותר, מכיוון שתמיד נוצרים תתי מודולים תוך כדי שלב הפיתוח עצמו, ויש צורך להכניס שינויים שלא הוגדרו בתכנון הראשוני. זה נובע מחוסר ניסיון בתכנון ופיתוח מערכת בסדר גודל כזה.

כרגע הפרויקט מוכן לגמרי כאשר יש לקחת בחשבון שתמיד יש אפשרות לשפר דברים שכבר נכתבו ולכן כל עוד המוצר לא יסופק ללקוח, יתבצעו בדיקות יזומות לייעול תהליכי המערכת ומניעת תקלות במידת האפשר.

b. פיתוחים עתידיים והמשך עבודה

פיתוחים עתידיים שאפשריים למערכת הם התממשקות של המערכת למערכות הבנק על מנת לייעל את העבודה ולמנוע מבקרי המוקד עבודה מיותר של הזנת נתוני האירועים למערכת, במידה וזה יתאפשר ניתן יהיה לבצע סינון אירועים אוטומטי לפי פרמטרים מוגדרים מראש, בצורה זו תופחת כמות האירועים שיגיעו למערכת ויינתן דגש רק לאירועים קריטיים שבהם צריך לטפל בזמן אמת.
בנוסף ניתן להרחיב את המערכת למערכת רבת משתמשים כך שכל עובדי המוקד יוכלו להשתמש בה.

c. רשימת מקורות

1. ירושלמי, אורי. (2003).***.NET & #C - מדריך מקצועי***. מרכז הדרכה 2000.
2. פרץ, שובל.(1998).***תכנון,ניתוח ועיצוב מערכות מידע, כרך א - תכנון***. תל אביב: האוניברסיטה הפתוחה.
3. פרץ, שובל.(1998).***תכנון,ניתוח ועיצוב מערכות מידע, כרך ב - ניתוח ועיצוב***. תל אביב: האוניברסיטה הפתוחה.
4. פרץ, שובל.(1998).***תכנון,ניתוח ועיצוב מערכות מידע, כרך ג - אב- טיפוס, כלי פיתוח, יישום וגישת העצמים***. תל אביב: האוניברסיטה הפתוחה.
5. Jones, Bradley L. (2002). ***#C - סדנת לימוד***.הרצליה: הוד- עמי.
6. Sharp, John.(2006).***Visual C# 2005 Step by Step***. Redmond, Washington: Microsoft Press.
7. Jesse, Liberty.(2005) ***Programming C#: Building .NET Applications with C#***
8. גלעד, ירון.(2006). "SIM/SOC - ניהול אבטחת מידע בעולם משתנה" DailyMaily, גיליון 4189.
9. הוראה 357 של בנק ישראל – ניהול טכנולוגיות מידע:

http://www.bankisrael.gov.il/deptdata/pikuah/nihul_takin/main.htm

נספח א - Software Requirements Document (SRD)

1. Introduction

1.1 Purpose

מטרת המסמך היא לתאר את המערכת מבחינת מטרות, פונקציונאלית תוך כדי הגדרת המודלים השונים של המערכת ופירוט דרישות המערכת למרכיביה השונים.

1.2 Scope of the software

תיחום המוצר הוא מערכת מידע לניהול, רישום ותיעוד אירועי אבטחת מידע שתבנה בהתאם לדרישות הלקוח.

1.3 Definitions, acronyms and abbreviations

DFD - Data Flow Diagram - תרשים זרימת נתונים
SQL - Structured Query Language - שפת שאילתות מובנית
GUI - Graphic User Interface - ממשק משתמש גראפי
SIM - Security Information Management - ניהול אבטחת מידע
SOC - Security operation Center - מוקד אבטחת מידע
DB - Data Base - בסיס נתונים
VB - Visual Basic - שפת תכנות

1.4 References

ספרות:

- פרץ, שובל.(1998).***תכנון,ניתוח ועיצוב מערכות מידע, כרך א - תכנון***. תל אביב: האוניברסיטה הפתוחה.
- פרץ, שובל.(1998).***תכנון,ניתוח ועיצוב מערכות מידע, כרך ב - ניתוח ועיצוב***. תל אביב: האוניברסיטה הפתוחה.
- גלעד, ירון.(2006). "SIM/SOC - ניהול אבטחת מידע בעולם משתנה" DailyMaily, גיליון 4189.
- ירושלמי, אורי. (2003).***C# & .NET - מדריך מקצועי***. מרכז הדרכה 2000.
- הוראה 357 של בנק ישראל – ניהול טכנולוגיות מידע:

http://www.bankisrael.gov.il/deptdata/pikuah/nihul_takin/main.htm

1.5. Overview of the document

מסמך SRD מתאר את אפיון המערכת.

בחלקו הראשון של הנספח מוצגות הגדרות מערכת, תיחום המערכת, מילון מונחים לשם הבנת המושגים השונים ורשימת מקורות להעשרת הידע.

בחלקו השני של הנספח מתוארות פונקציונאליות ומטרות המערכת, תנאים שבהם תפעל המערכת ומי יפעיל אותה. דברים נוספים שנסקרים בחלקו השני של הנספח הם: אילוצי המערכת, התממשקות למערכות אחרות, והצגת מודל המערכת באמצעות תרשימי זרימה ו ERD כאשר זהו החלק המרכזי והחשוב של הנספח כולו.

הצגת מודל המערכת מתוארת ע"י התהליכים השונים הקיימים במערכת (תרשימי זרימה) והצגת תרשים קשרי הישויות (ERD) המתאר את מבנה בסיס הנתונים של המערכת.

חשוב לציין שחל שינוי באפיון המערכת בשלב הפיתוח לכן יש הבדל בין המודולים הראשוניים שהוגדרו בנספח זה (בזמן הגשת דו"ח ביניים 1) למודולים הקיימים במערכת.

במסמך הראשוני הוגדרו 5 מודולים שהם:

1.1.1. מודל הרשאות גישה למערכת.
1.1.2. מודל תיעוד וסיווג אירועים + אופן טיפול בכל אירוע.
1.1.3. מודל חיפוש אירועים במאגר אירועים חריגים.
1.1.4. מודל הפקת דו"חות אירועים.
1.1.5. מודל לניהול פורטל ידע.

במערכת הסופית ישנם 4 מודולים מרכזיים (מעודכנים בנספח) שהם:

1. **מודול לניהול אירועים שמכיל בתוכו את מודולים 2-4 שהוגדר בנספח SRD.**
2. **מודול לניהול ידע שמכיל את מודול 5 שהוגדר בנספח SRD.**
3. **מודול ניהול משתמשים שהוא מודל חדש שהתווסף בזמן פיתוח ולא הוגדר במסמך זה. הוא מכיל בין היתר את מודול 1 שהוגדר בנספח SRD.**
4. **מודול ניהול משמרות שהתווסף בזמן פיתוח המערכת.**

בחלקו השלישי של הנספח מתוארות דרישות פונקציונאלית, דרישות ביצועים, דרישות ממשק, המשאבים הנדרשים לתפעול המערכת, דרישות בדיקה ודרישות אבטחה. החלק השלישי הוא מאוד חשוב מכיוון שבאמצעות הדרישות השונות ניתן לאפיין בצורה הכי טובה את המערכת העתידית.

2. General Description

2.1. Relation to current projects - N/A

2.2. Relation to predecessor and successor projects - N/A

2.3. Function and purpose

מערכת המידע מיועדת עבור ריכוז וניהול אירועים חריגים ע"י בקרי מוקד אבטחת המידע. המערכת שתבנה כחלק מהפרויקט תשמש את המוקד בעבודה השוטפת של תיעוד אירועים והפקת הדו"חות עבור אירועי אבטחת מידע חריגים, כמו כן המערכת תכיל נוהל טיפול עבור כל אירוע חריג שאירע באחת המערכות. בנוסף המערכת תאפשר הפקת דוחות פנימיים של המוקד, כגון: דו"ח חפיפה בחילוף המשמרות בין הבקרים במוקד ודו"ח מרוכז של אירועים חריגים לפי סינון רלוונטי עבור מנהל המוקד. דבר נוסף שיהיה במערכת הוא פורטל לניהול ידע שיכיל את כל הנהלים הרלוונטיים לעבודה השוטפת.
ניתן לראות את תרשימי הזרימה שמתארים את פונקציונאליות המערכת בסעיף 2.7.

2.4. Environmental considerations

המערכת מיועדת לעבודה השוטפת של מוקד אבטחת מידע הוא חלק ממחלקת אבטחת המידע שהיא אחת ממחלקות המטה של בנק דיסקונט. ניתן לראות את מבנה הארגון בצורה מפורטת בפרק 2 של נספח ד - הצעת פרויקט מורחבת.

הגורמים המעורבים בהכנת הפרויקט

לקוח מטעם הבנק: מאיר סלר - מנהל מוקד אבטחת מידע, בנק דיסקונט.

מנחה הפרויקט: גב' יונית שוורץ וורבר - פרויקטורית במגמה להנדסת תוכנה.

מפתח הפרויקט: ויקטור קייטמזוב.

המשתמשים של המערכת הם בקרי המוקד, אנליסט המוקד ומנהל המוקד כאשר לכל סוג משתמש יהיו הרשאות גישה שונות למערכת.
תפעול ותחזוקת המערכת תתבצע ע"י הבקרים ואנליסט המוקד.

ציוד חומרה ותוכנה בסביבת הרצה (צד לקוח)

תוכנות:

- מערכת הפעלה מסוג Windows 2000 ומעלה עם עדיפות למערכת הפעלה Windows XP Professional.
- תוכנות Microsoft Office.
- Microsoft SQL Server 2005 (עבור בסיס הנתונים).

חומרה: עבור מחשב או רשת מחשבים שמריצות את מערכת המידע:

- מעבד פנטיום 4.
- זיכרון 256MB ומעלה.
- כרטיס רשת.
- כבל רשת.

2.5. Relation to other systems

מערכת המידע מהווה אפליקציה עצמאית בלתי תלויה במערכות אחרות כלומר מערכת המידע שתבנה עבור המוקד לא תתממשק למערכות הבנק השונות אלא תשמש לניהול העבודה השוטפת של המוקד בלבד.

2.6. General constraints

אילוצי זמן:

המערכת לא תתממשק למערכות הבנק.
הסיבה: זמן פיתוח ארוך מדי.

התאמת המערכת למשתמשים:

עיצוב המערכת יהיה ידידות ונוח לשימוש
הסיבה: רוב עובדי המוקד הם ללא הרבה ניסיון בתחום התוכנה, רובם סטודנטים בשנים הראשונות, סיבה נוספת היא שבקרם עובדים עם המערכת מסביב לשעון ולכן חשוב שהבקרים לא יתעייפו לעבוד עם המערכת במקרה שהמערכת קשה להפעלה.

אילוצי עלות:

המערכת תרוץ בשלה הראשוני על תחנה בודדת עם בסיס נתונים מקומי:
הסיבה: חסכון בתקציב לרכישת ציוד מחשבים

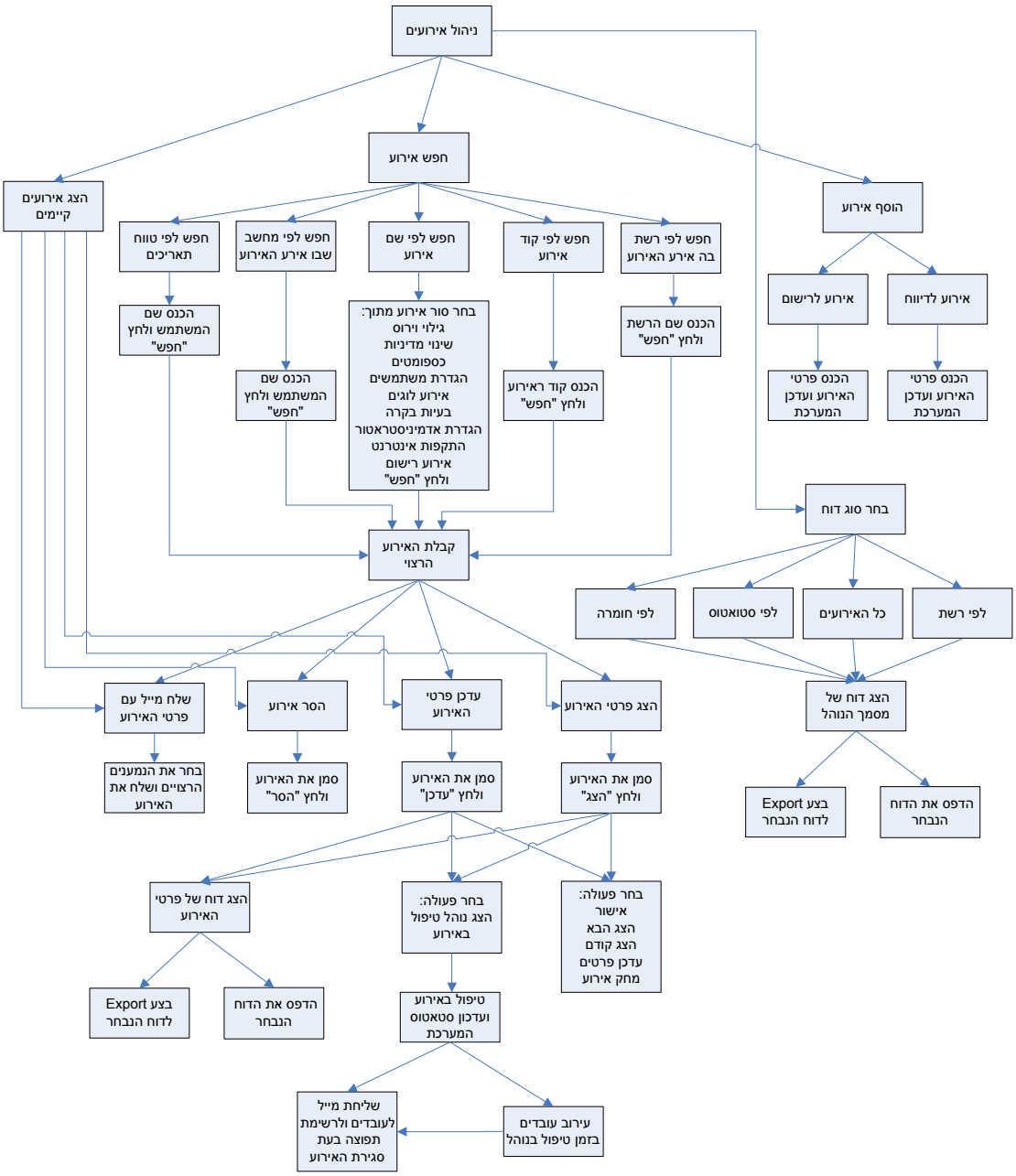
2.7. Model description

במערכת קיימים 4 מודולים מרכזיים:

1. ניהול אירועים
2. ניהול משתמשים
3. ניהול ידע
4. ניהול משמרות

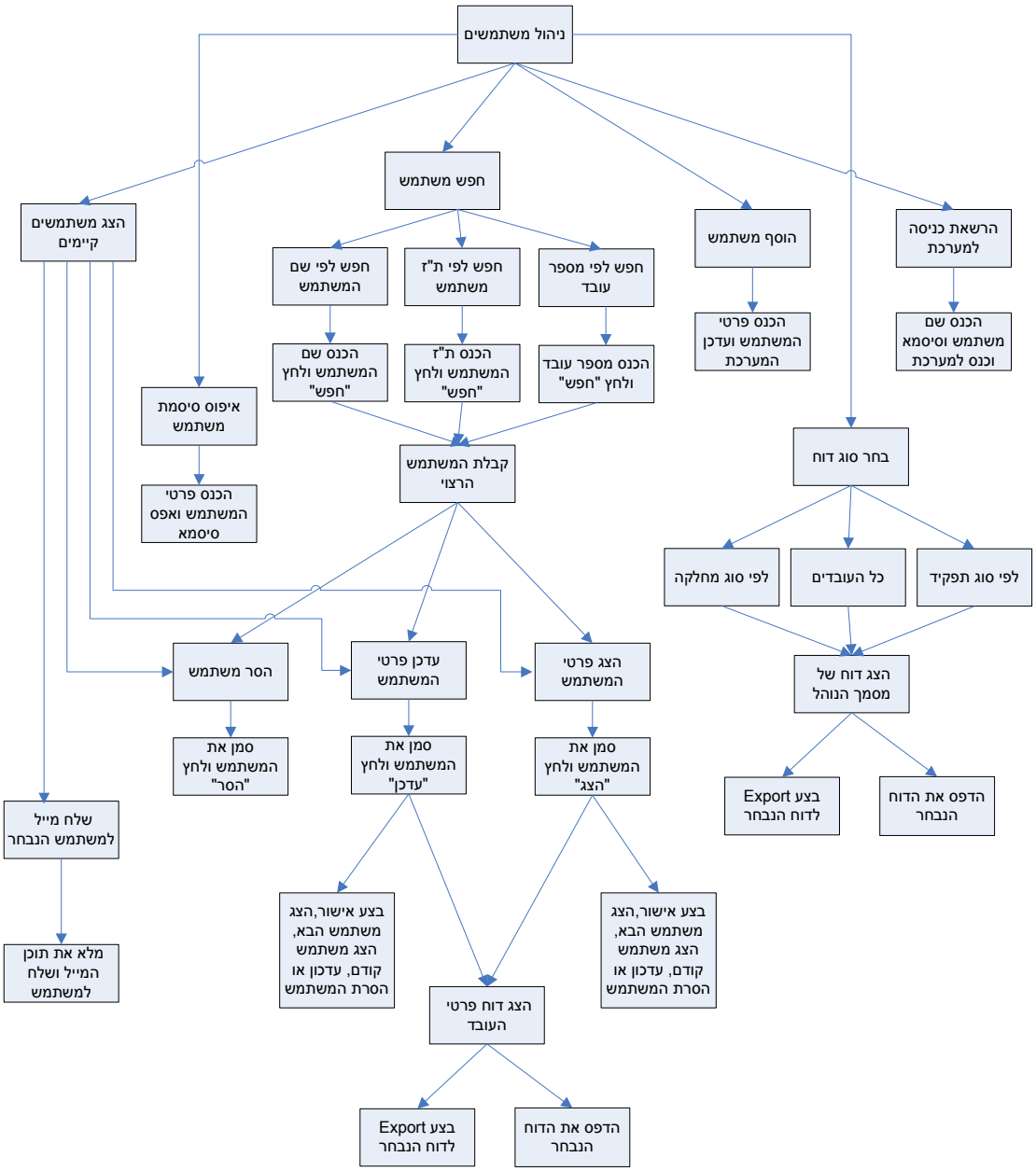
ניתוח תהליכי המערכת מיוצגים ע"י תרשימי זרימה פונקציונאליים:

מודל ניהול אירועים



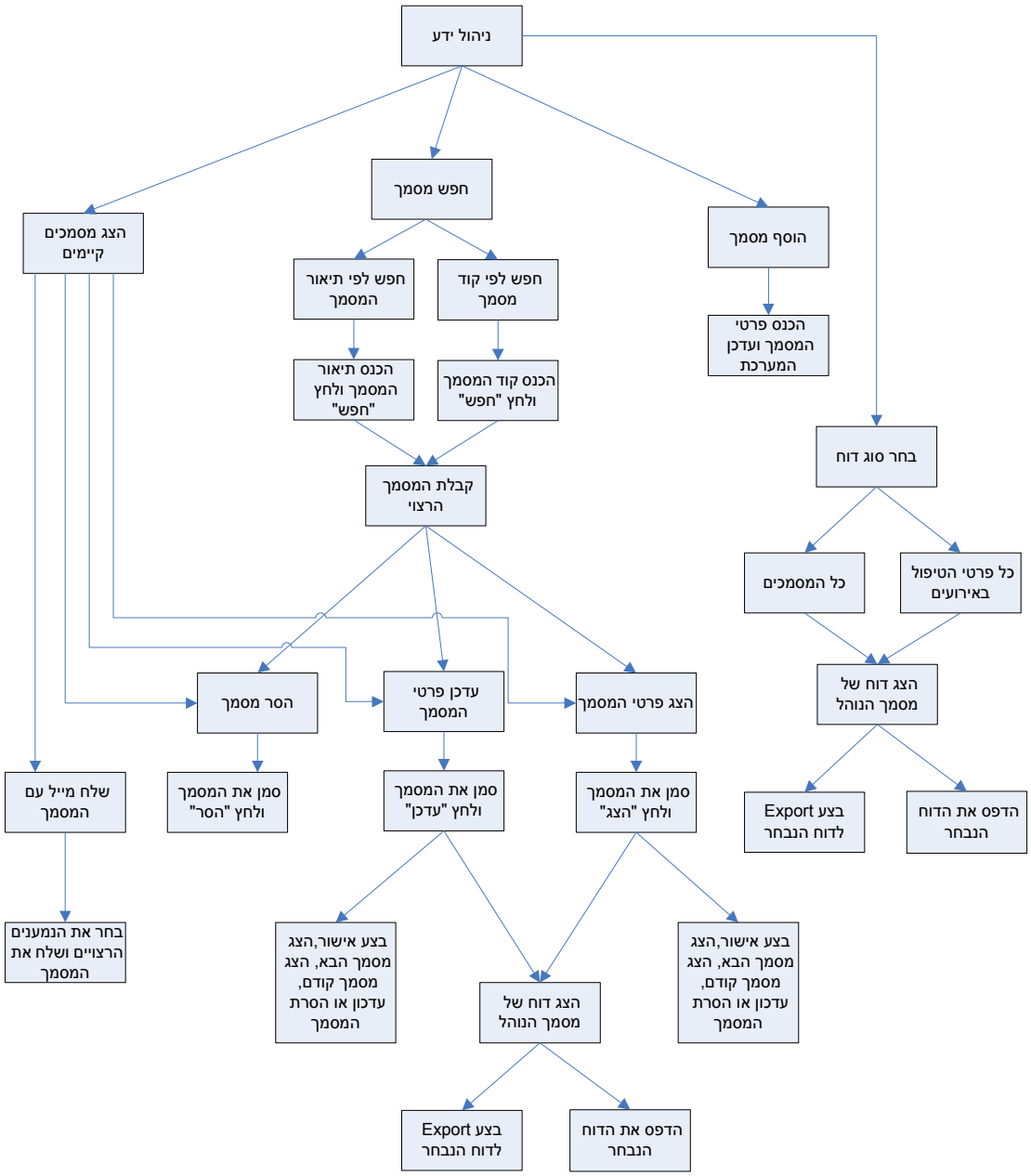
איור 1 – תרשים מודול ניהול אירועים

מודל ניהול משתמשים



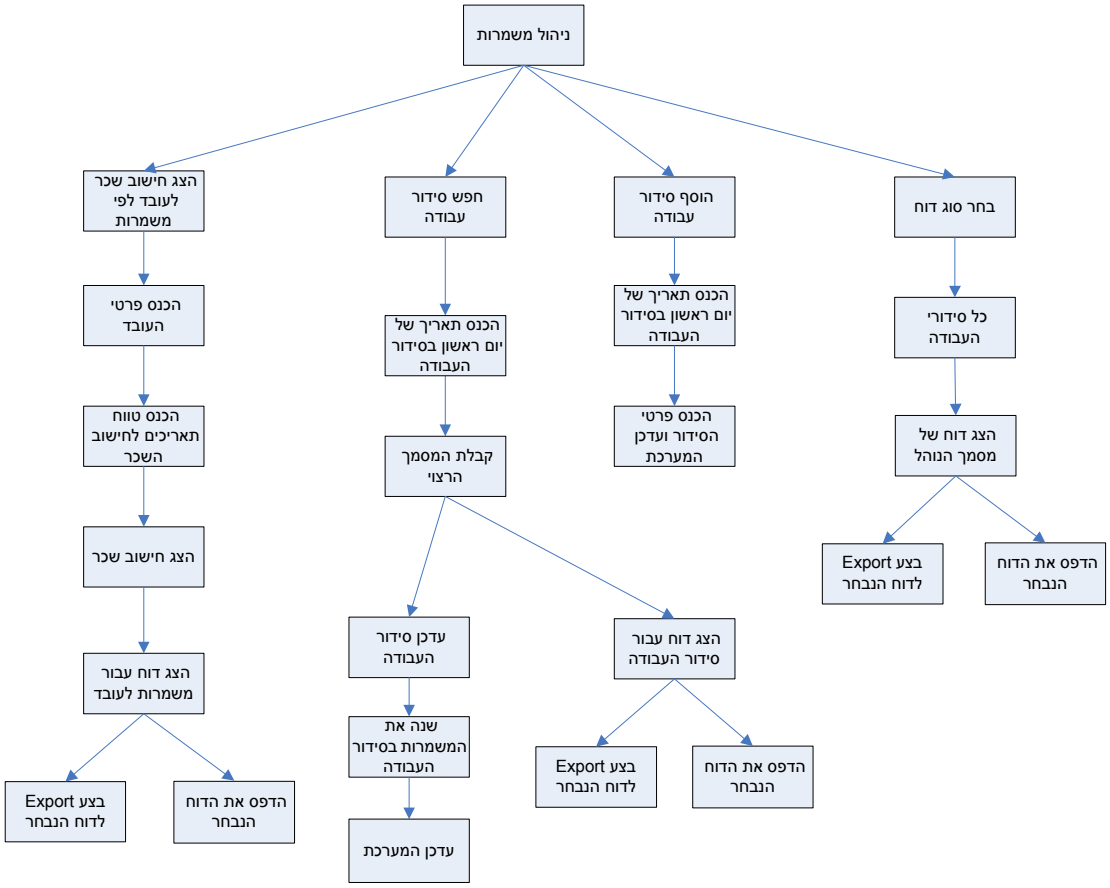
איור 2 – תרשים מודול ניהול משתמשים

מודל ניהול ידע



איור 3 – תרשים מודול ניהול ידע

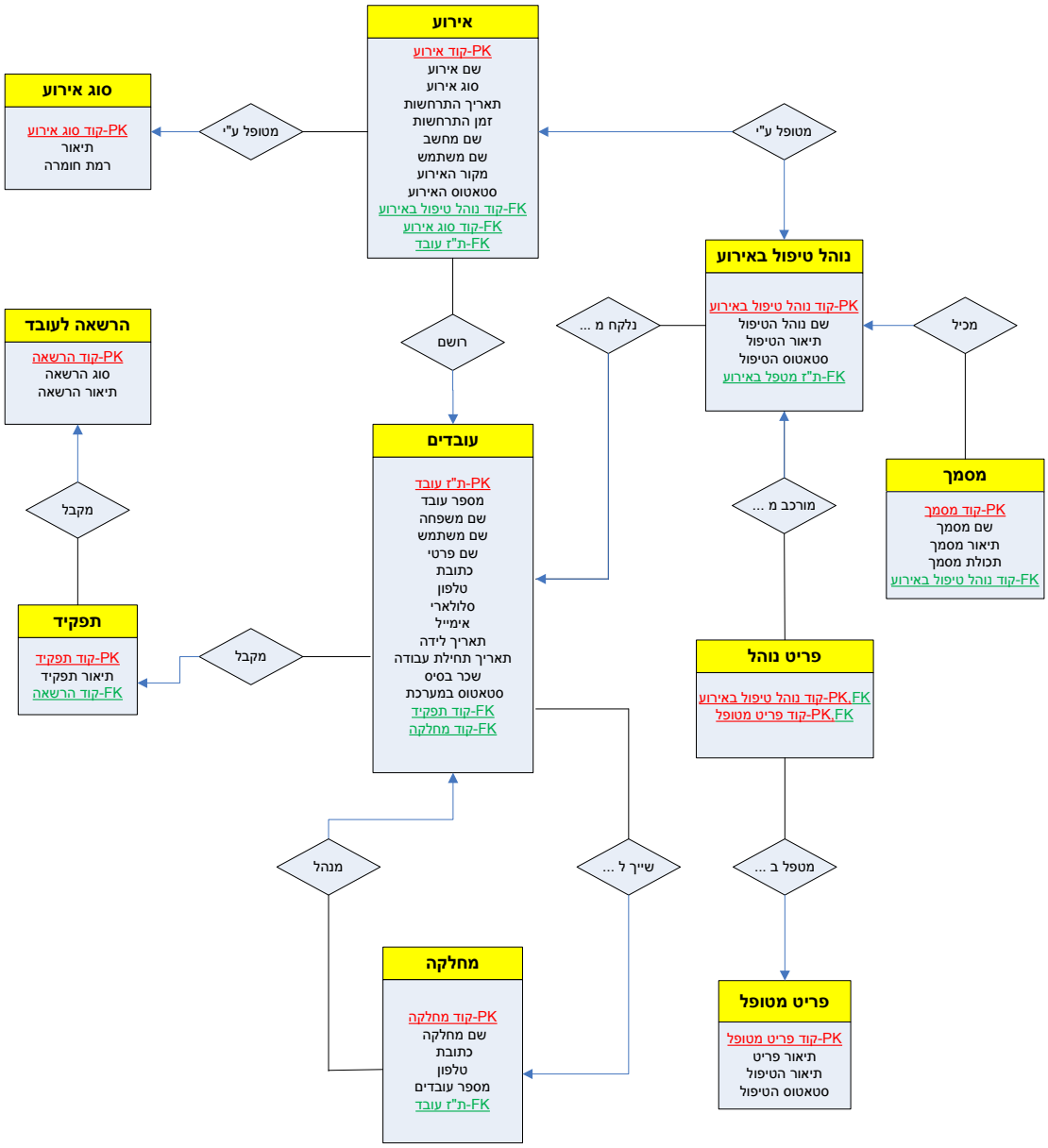
מודל ניהול משמרות



איור 4 – תרשים מודול ניהול משמרות

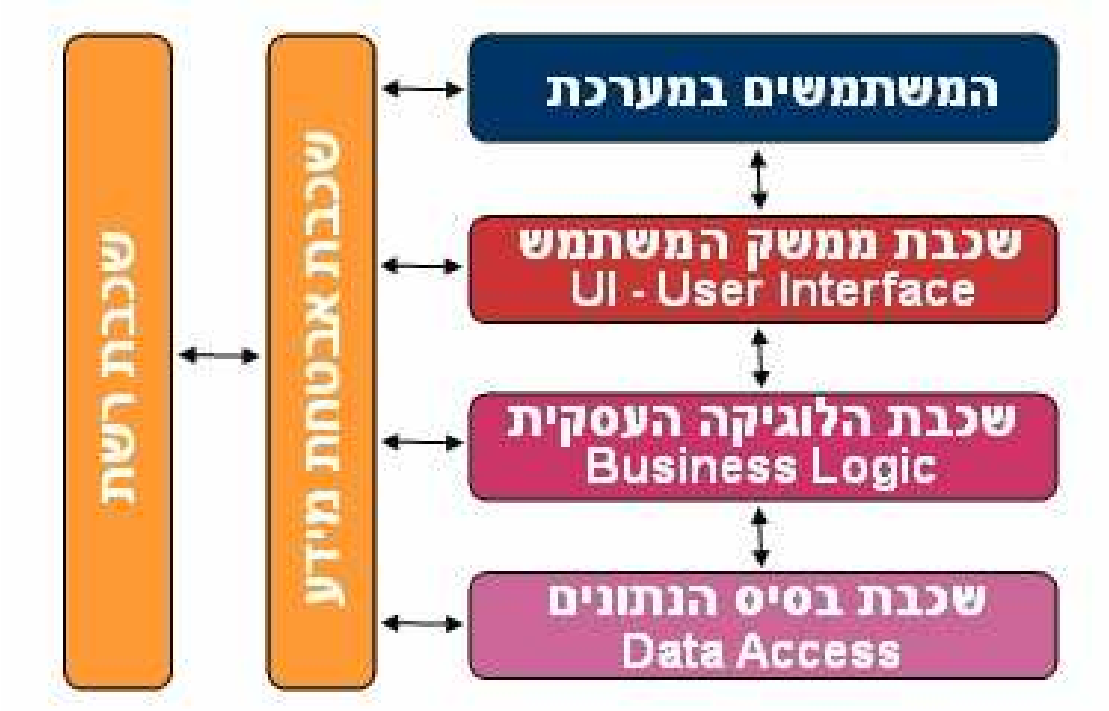
תרשים ERD עבור בסיס הנתונים של הפרויקט

ERD



איור 5 – תרשים ERD של הפרויקט

תרשים הארכיטקטורה של המערכת:



איור 6 – תרשים ארכיטקטורת המערכת

מערכת המידע תהיה מורכבת מ 4 שכבות עיקרית:

1. שכבת ממשק המשתמש הגראפי שתקשר בין המערכת למשתמש.
2. שכבת הלוגיקה העסקית שתבצע את כל הפעולות הלוגיות של מערכת המידע.
3. שכבת בסיסי הנתונים שתכיל את מאגרי המידע של המערכת.

בנוסף במידה ונרצה להרחיב את המערכת לאפליקציה רשתית, נכניס שכבה נוספת של אבטחת מידע בין הרשת לכל אחד מהשכבות של המחשב המריץ את מערכת המידע.

3. Specific Requirements

3.1. Functional requirements - דרישות פונקציונאליות

- רישום אירוע חריג במערכת.
- הגדרת סטאטוס עבור כל אירוע והצגת אופן הטיפול בו.
- הגדרת סטאטוס אירועים חריגים בזמן אמת.
- הגדרת אופן טיפול באירוע שנרשם במערכת.
- חיפוש אירוע לפי פרמטר ספציפי.
- הגדרת תבנית דו"ח מיוחדת לכל סוג של אירוע אבטחת מידע חריג.
- סינון והפקת דו"חות אירועים חריגים.
- הפקת דו"חות חפיפה בין משמרות הבקרים במוקד.
- ביצוע מעקב אחרי דו"חות החפיפה של המערכת.
- הצגת התראה על אירוע חריג שחוזר על עצמו במערכת.
- קליטת עובד חדש במוקד ועדכונו כמשתמש מערכת.
- ניהול פורטל ידע שירכז את כל הנהלים הרלוונטיים לעבודה השוטפת.
- עדכון מסמכי נהלים בפורטל הידע.

3.2. Performance requirements

המערכת תרוץ בשלב הראשוני במחשב אחד עם בסיס נתונים מקומי כאשר זמן התגובה של הפעולות יהיה לפי הזמנים הבאים:

עדכון פעולת רישום אירוע: עד 5 שניות.
שאילתות מערכת: עד 10 שניות בהתאם לגודל השאילתא.
העלאת מסך טיפול באירוע: עד 10 שניות.
הפקת דו"חות: עד דקה בהתאם לגודל הדו"ח.

המקום הדרוש לבסיס הנתונים יגדל עם הזמן ולא צפויים עומסים מיוחדים על המערכת.

3.3. Interface requirements

מערכת המידע לניהול, תיעוד ורישום אירועי אבטחת מידע תבנה כאפליקציית Win-Form ותתחבר לבסיס הנתונים שימוקם על אותו מחשב בו מותקנת האפליקציה.
מבנה הממשק של המערכת יכיל מסך ראשי שמתוכו יהיה ניתן לעשות את כל הפעולות בסיסיות של הוספת אירועים, טיפול באירוע פתוח, הצגת דו"חות שונים ועוד. כמו כן המסך הראשי יכיל ניתוח של מצב האירועים החריגים במוקד בזמן אמת. כאשר מבצעים פעולה מסוימת מהחלון הראשי, נפתח חלון חדש שעליו ממשיכים לבצע את הפעולה הרצויה.
המערכת תפעל כ - Stand Alone ולא תתממשק עם מערכות הבנק השונות.

3.4. Resource requirements - תיאור סביבת פיתוח

כלים:

- MS-Project.
- UML (Use Case, Class Diagram, Sequence Diagram).
- Microsoft Office – Visio (לצורך תרשימי DFD להגדרת המערכת).

תוכנות:

- מערכת הפעלה מסוג Windows XP Professional.
- תוכנות Microsoft Office עם עדיפות לגרסאות 2003/ 2007.
- Microsoft Visual Studio® .NET 2005 (Framework 2.0)
- Microsoft SQL Server 2005 (עבור בסיס הנתונים).

חומרה: עבור מחשב שמפתח את המערכת המידע:

- מעבד פנטיום 4.
- זיכרון 512MB ומעלה.

3.5. Verification requirements - אימות דרישות

סביבת הבדיקות שתבדוק את תפקוד המערכת שנבנתה היא מחשב עם בסיס נתונים מקומי שיריץ את האפליקציה של מערכת המידע , כמו כן ניתן להציג סימולציה של מספר מחשבים שמחוברים ברשת למחשב (שרת) שיכיל את בסיס הנתונים של מערכת המידע.
אין שום צורך לחבר את המחשבים שיריצו את האפליקציה למערכות הבנק, זאת עקב העובדה שהמערכת שתבנה תשמש את מוקד אבטחת המידע בלבד.

כמו כן חשוב לציין שמערכת המידע שתבנה בפרויקט לא תתממשק למערכות הבנק עקב העובדה שדבר זה כרוך באישורי הנהלה רבים, בהשקעה רבה מאוד בבניית המערכת ובצורך לבצע שינויים חלקיים במערכות הקיימות של הבנק.

3.6. Security requirements - דרישות בטחון ואבטחה

נושא אבטחת המידע הוא נושא מאוד חשוב בכל מה שקשור לעבודה השוטפת של הבנק בכלל ומחלקת אבטחת המידע בפרט, לכן חייבים לאבטח את המערכת שתבנה בפרויקט עקב הסיבה שהמערכת תכיל חומר מאוד רגיש הכולל בין היתר מסמכים המכילים פעולות בנקאיות, פרטי לקוחות, שמות משתמשים ומחשבים השייכים לרשת המטה ולרשת הסניפים ועוד, לכן יש צורך לאבטח את המידע של המערכת בצורה הבאה:

- לכל משתמש במערכת יהיה שם משתמש וסיסמא ייחודיים כדי למנוע כניסה של אנשים שלא מורשים לגשת למערכת.
- לכל משתמש יהיה סוג הרשאה המתאימה לתפקיד.
- חל איסור מוחלט להוציא דו"חות מערכת מחוץ לתחומי העבודה כאשר תחום העבודה זה בניין ההנהלה של בנק דיסקונט.

האבטחה שהבנק יספק למחשבים שבהם תרוץ מערכת המידע היא אבטחה בסיסית של שם משתמש וסיסמא בכניסה למערכת ההפעלה של המחשב, וזאת עקב העובדה שיש רשימת עובדים מורשים מוגדרת שיכולים להיכנס למוקד ולבוא במגע עם המערכות השונות בו, שזה גם סוג של אבטחה פיזית.

נספח ב - Software Design Description (SDD)

1. Introduction

1.1 Purpose

מטרת המסמך היא לתאר את ארכיטקטורת המערכת, להגדיר קומפוננטות מרכזיות במערכת ולהציג ממשק גראפי למשתמש תוך כדי הגדרת המודלים השונים של המערכת ופירוט פונקציונאליות המערכת למרכיביה השונים.

1.2 Scope of the software

תיחום המוצר הוא מערכת מידע לניהול, רישום ותיעוד אירועי אבטחת מידע שתבנה בהתאם לדרישות הלקוח.

1.3 Definitions, acronyms and abbreviations

DFD - Data Flow Diagram - תרשים זרימת נתונים
SQL - Structured Query Language - שפת שאילתות מובנית
GUI - Graphic User Interface - ממשק משתמש גראפי
SIM - Security Information Management - ניהול אבטחת מידע
SOC - Security operation Center - מוקד אבטחת מידע
DB - Data Base - בסיס נתונים

1.4 References

10. פרץ, שובל.(1998).***תכנון,ניתוח ועיצוב מערכות מידע, כרך א - תכנון***. תל אביב: האוניברסיטה הפתוחה.

11. פרץ, שובל.(1998).***תכנון,ניתוח ועיצוב מערכות מידע, כרך ב - ניתוח ועיצוב***. תל אביב: האוניברסיטה הפתוחה.

12. פרץ, שובל.(1998).***תכנון,ניתוח ועיצוב מערכות מידע, כרך ג - אב- טיפוס, כלי פיתוח, יישום וגישת העצמים***. תל אביב: האוניברסיטה הפתוחה.

1.5 Overview of the document

מסמך SDD מתאר את ארכיטקטורת המערכת.

בחלקו הראשון של הנספח מוצגות הגדרות מערכת, תיחום המערכת, מילון מונחים לשם הבנת המושגים השונים ורשימת מקורות להעשרת הידע.

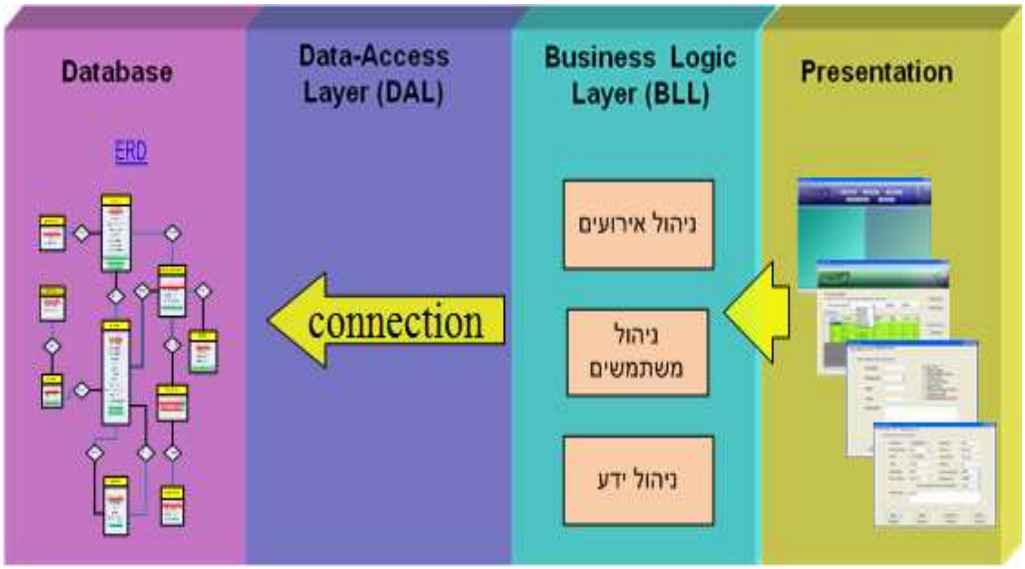
בחלקו השני של הנספח מתוארות ארכיטקטורת המערכת, והמודולים המרכזיים שמגדירים את ממשק המערכת. הצגת ארכיטקטורת המערכת מתוארת באמצעות תרשים כאשר לכל מודול מרכזי במערכת יש תרשים מפורט של פונקציונאליות הן מבחינת השכבה הלוגית והן מבחינת ממשק המשתמש.

בחלקו השלישי של הנספח מתוארות הקומפוננטות המרכזיות של המערכת, ובחלקו הרביעי מתואר בצורה מפורטת הממשק הגראפי למשתמש.

2. System Architectural Design

2.1. Chosen System Architecture

תיכון המערכת יתבצע בשיטת השכבות בצורה המתוארת באיור:



איור 1 – תיכון המערכת בשיטת השכבות

המערכת תתחלק ל 4 שכבות הבאות:

Presentation Layer – שכבה האחראית על הצגת ממשק גרפי למשתמש.

Business Logic Layer – שכבה האחראית על פעולות לוגיות של התוכנה, מכילה מחלקות המייצגות את הישויות המרכזיות במערכת ואחראית על הפעלת שאילתות לצורך עבודה מול בסיס הנתונים (עבודה מול בסיס הנתונים יעשה באמצעות שאילתות דינאמיות ובאמצעות stored procedures).

Data Access Layer – שכבה האחראית על יצירת תקשורת בין שכבת הלוגיקה לשכבת בסיס הנתונים.

Database – שכבה המכילה את נתוני הטבלאות בבסיס הנתונים ופרוצדורות שמורות שמופעלות משכבת הלוגיקה.

2.2. System Interface Description

External (התממשקות חיצונית)

מערכת המידע מהווה אפליקציה עצמאית בלתי תלויה במערכות אחרות כלומר מערכת המידע שתבנה עבור המוקד תפעל כ - Stand Alone ולא תתממשק למערכות הבנק השונות אלא תשמש לניהול העבודה השוטפת של המוקד בלבד.

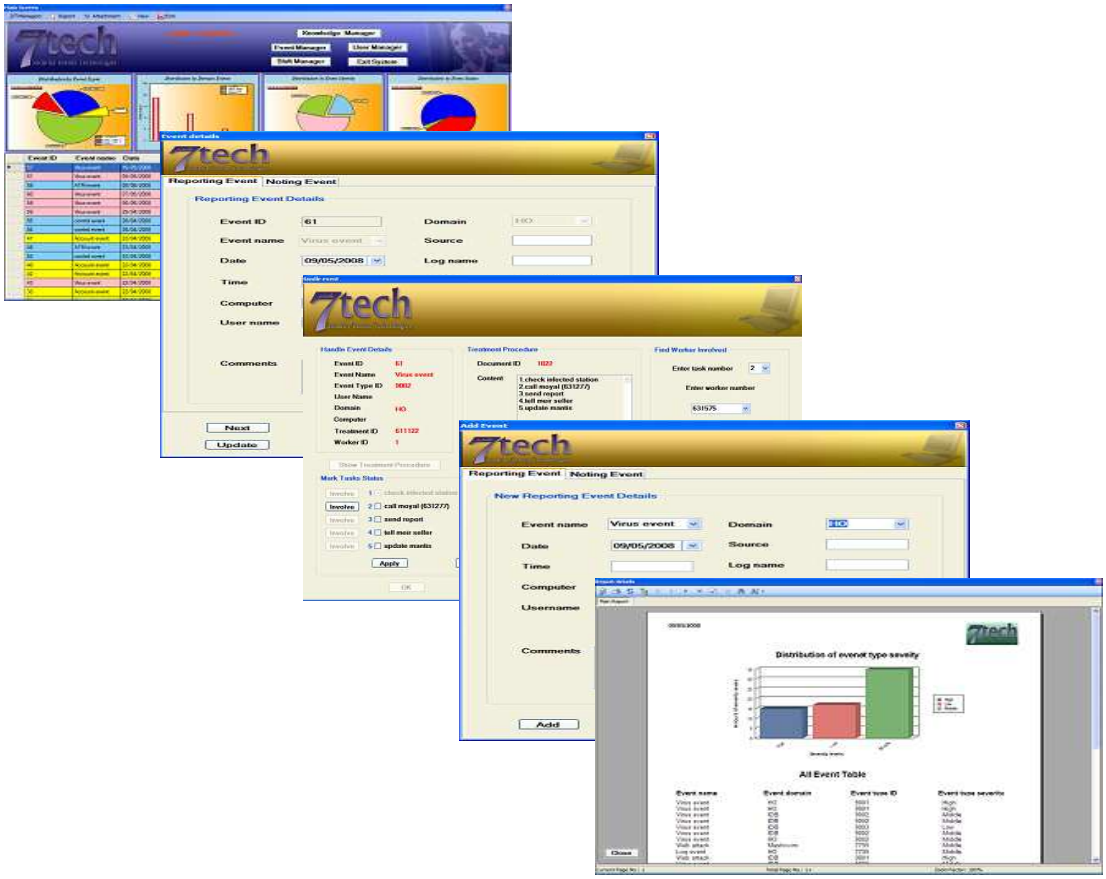
Internal (התממשקות פנימית)

ממשק המערכת מתחלק ל 4 מודולים מרכזיים:
1) ניהול אירועים
2) ניהול משתמשים
3) ניהול ידע
4) ניהול משמרות

כעת נתאר כיצד בא לידי ביטוי כל אחד מהמודולים המרכזיים בשכבת ה GUI ובשכבת הלוגיקה:

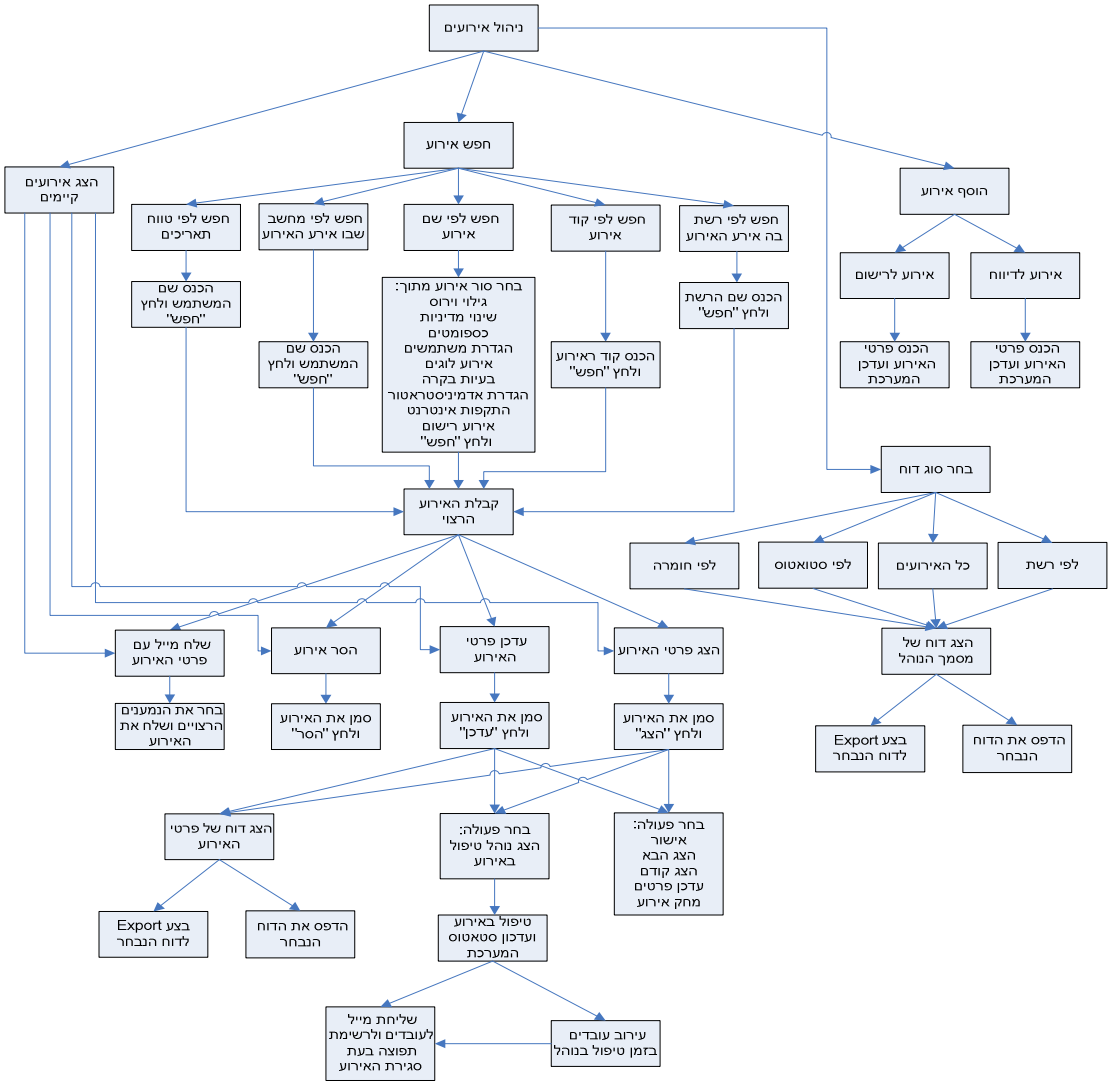
2.2.1 ניהול אירועים:

Presentation Layer:



איור 2 – ממשק גראפי של מודול ניהול האירועים

Business Logic Layer: ניהול אירועים



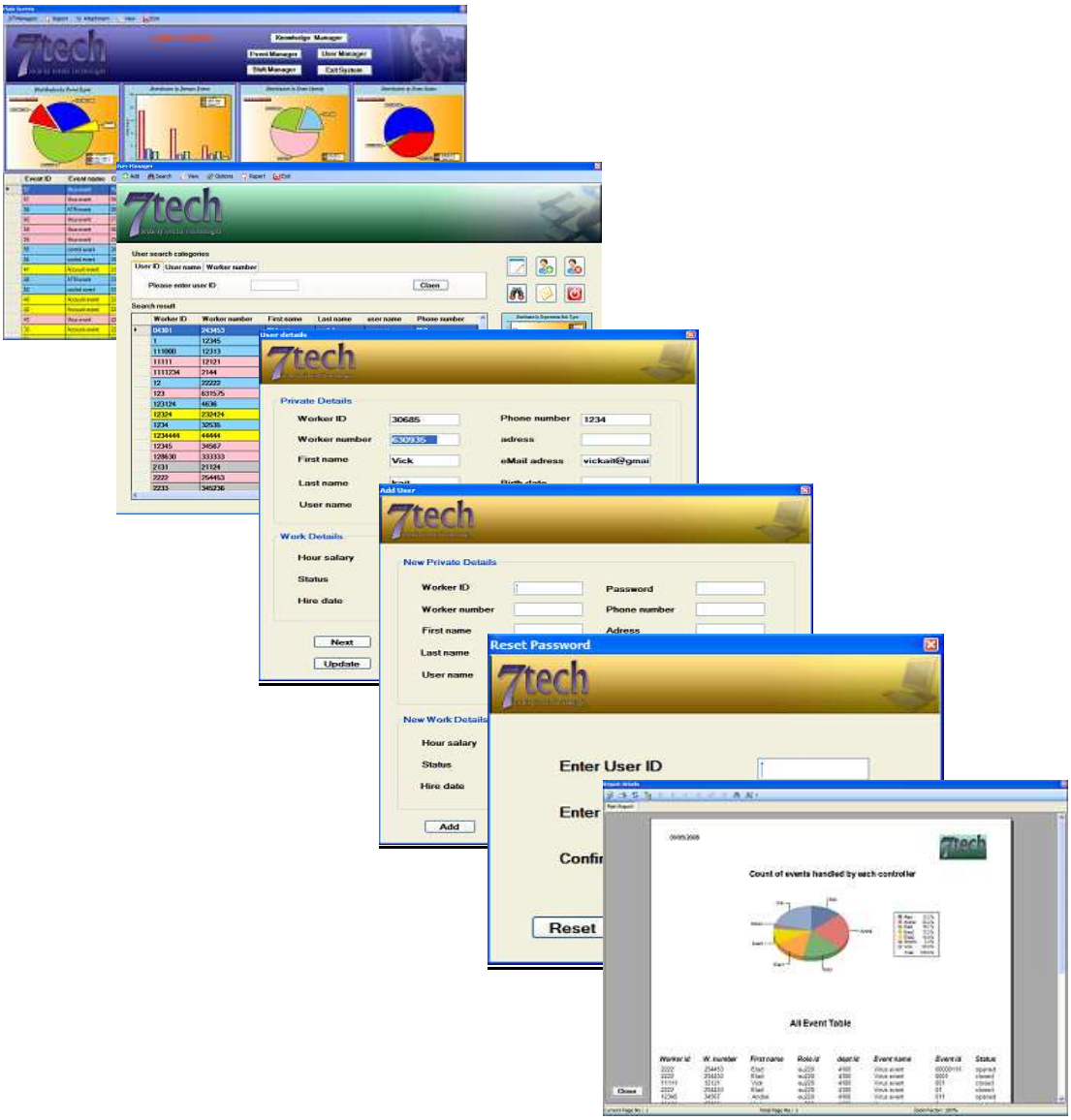
איור 3 – פונקציונאליות בשכבת הלוגיקה של מודול ניהול האירועים

מודול ניהול האירועים (איור 2 + איור 3) מאפשר לבצע את הפעולות הבאות:

- הקמת משתמש חדש ושמירתו במערכת
- הצפנת סיסמת משתמש בעת הוספת משתמש לבסיס הנתונים
- שליפת פרטי משתמש לצורך עדכון במערכת
- ביצוע חיפוש דינמי לפי פרמטרי המשתמש
- הצגה גראפית של התפלגות המשתמשים לפי סוגי המחלקות וחלוקת התפקידים בכל מחלקה
- חיפוש ושליפת פרטי משתמש בעת טיפול באירוע חריג לצורך תיוק עובד לפריט ספציפי וביצוע מעקב על העובדים המעורבים בטיפול
- אפשרות ביצוע איפוס סיסמא למשתמשים השונים
- שליחת מייל לעובד הרלוונטי
- הפקת דו"חות על המשתמשים השונים במערכת

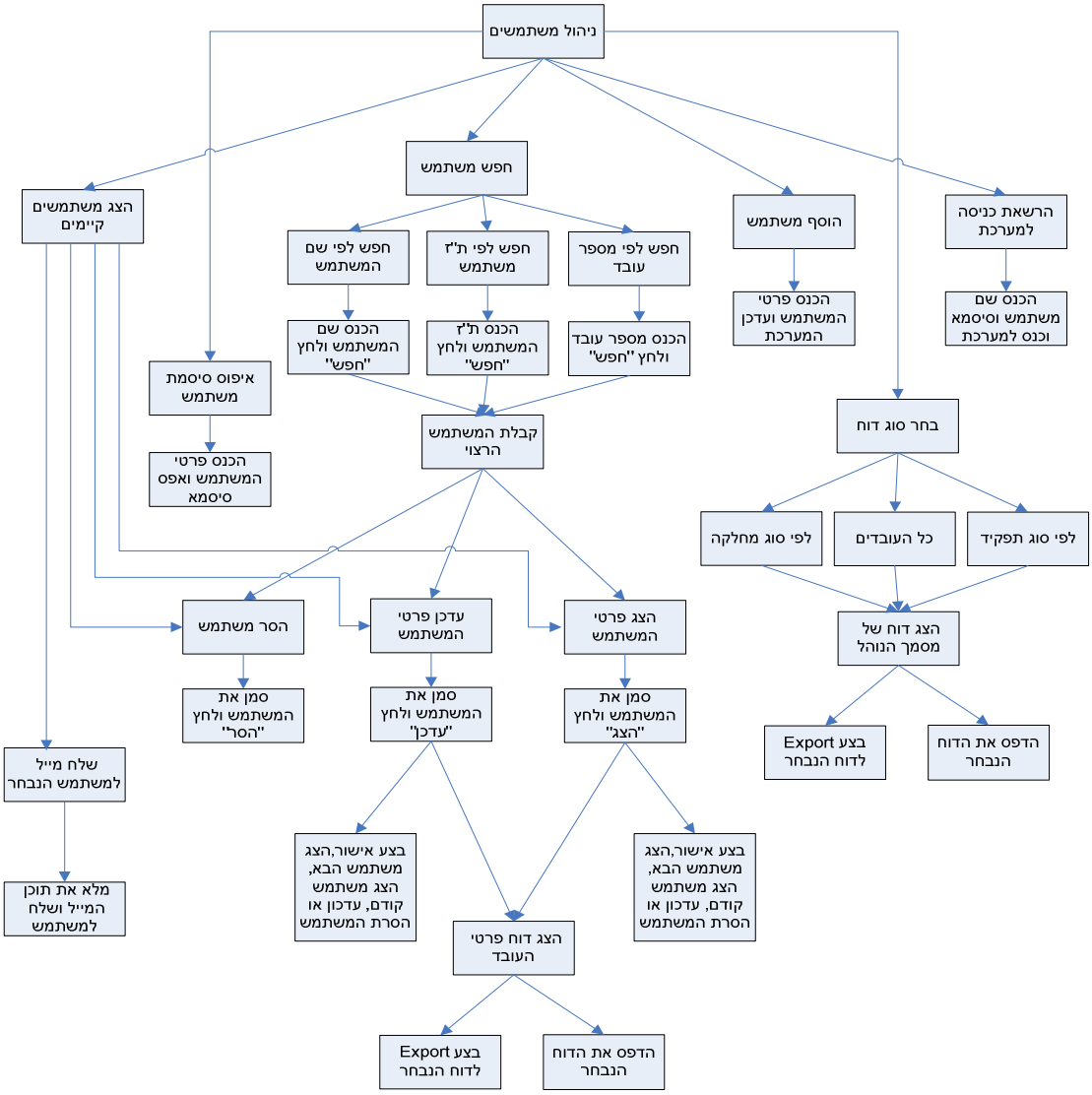
2.2.2 ניהול משתמשים:

Presentation Layer:



איור 4 – ממשק גראפי של מודול ניהול המשתמשים

Business Logic Layer: ניהול משתמשים



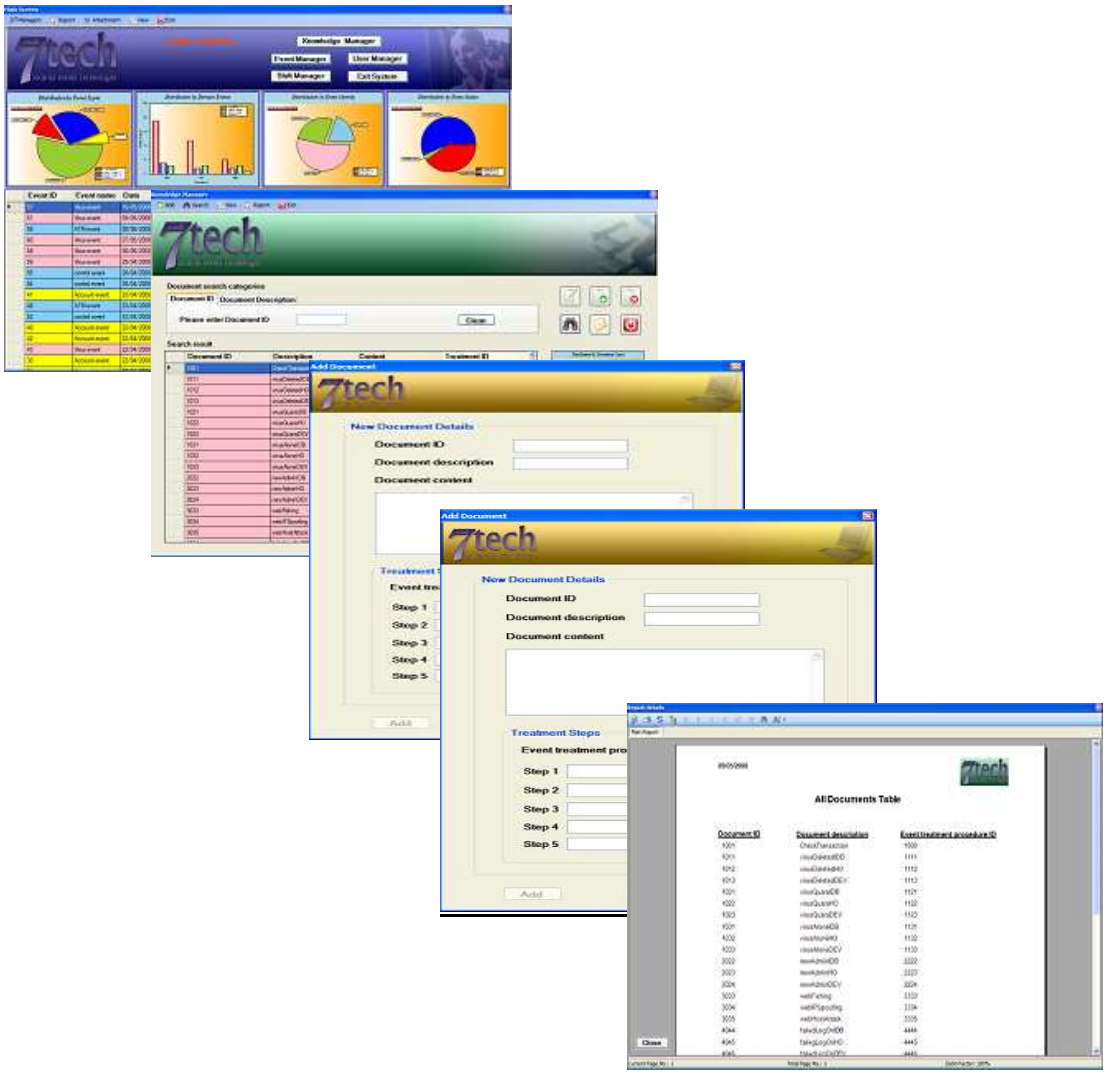
איור 5 – פונקציונאליות בשכבת הלוגיקה של מודול ניהול המשתמשים

מודול ניהול המשתמשים (איור4 + איור 5) מאפשר לבצע את הפעולות הבאות:

- הקמת משתמש חדש ושמירתו במערכת
- הצפנת סיסמת משתמש בעת הוספת משתמש לבסיס הנתונים
- שליפת פרטי משתמש לצורך עדכון במערכת
- ביצוע חיפוש דינמי לפי פרמטרי המשתמש
- הצגה גראפית של התפלגות המשתמשים לפי סוגי המחלקות וחלוקת התפקידים בכל מחלקה
- חיפוש ושליפת פרטי משתמש בעת טיפול באירוע חריג לצורך תיוק עובד לפריט ספציפי וביצוע מעקב על העובדים המעורבים בטיפול
- אפשרות ביצוע איפוס סיסמא למשתמשים השונים
- שליחת מייל לעובד הרלוונטי
- הפקת דו"חות על המשתמשים השונים במערכת

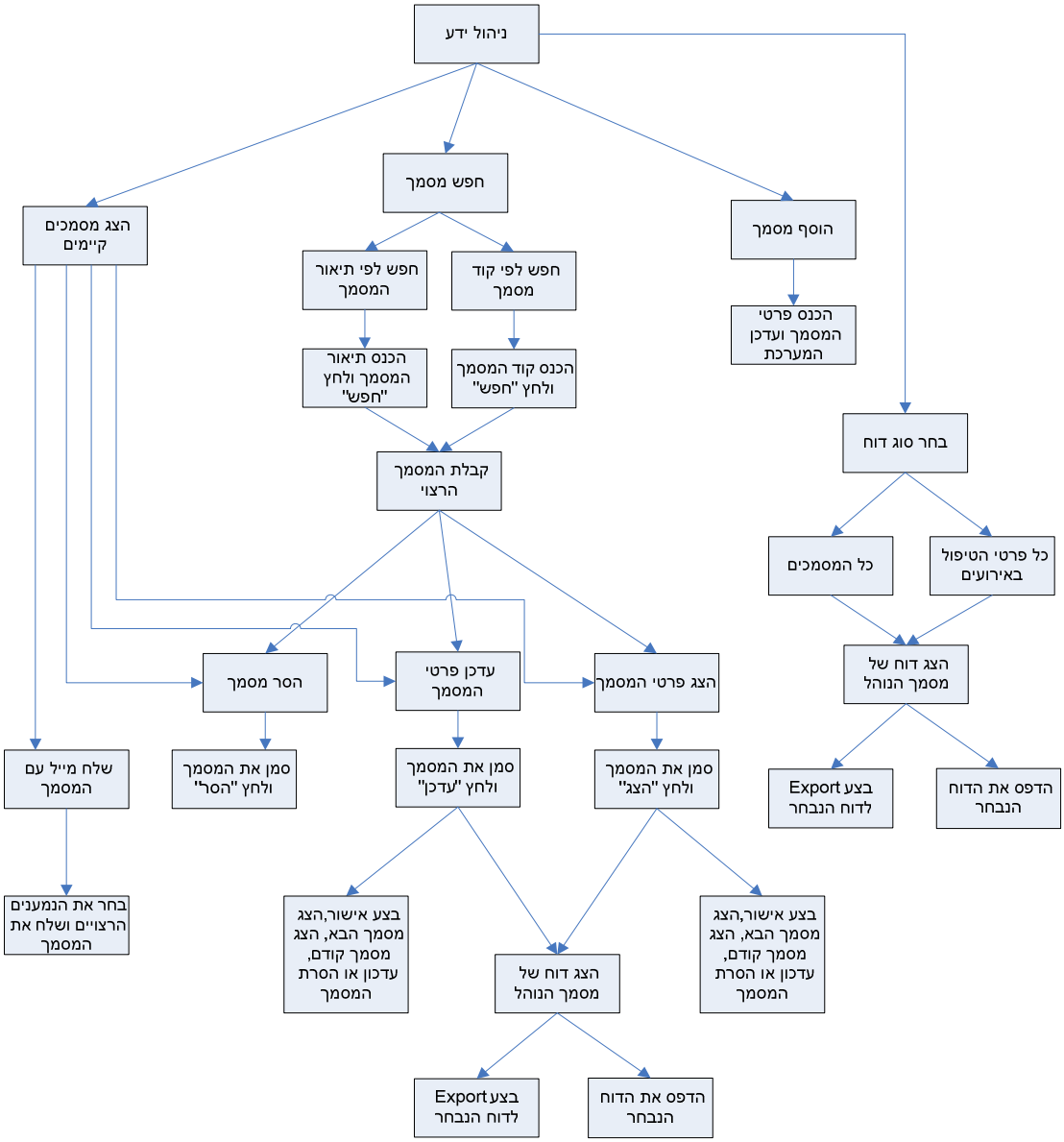
2.2.3 ניהול ידע:

Presentation Layer:



איור 6 – ממשק גראפי של מודול ניהול הידע

Business Logic Layer: ניהול ידע



איור 7 – פונקציונאליות בשכבת הלוגיקה של מודול ניהול הידע

מודול ניהול הידע (איור 6 + איור 7) מאפשר לבצע את הפעולות הבאות:

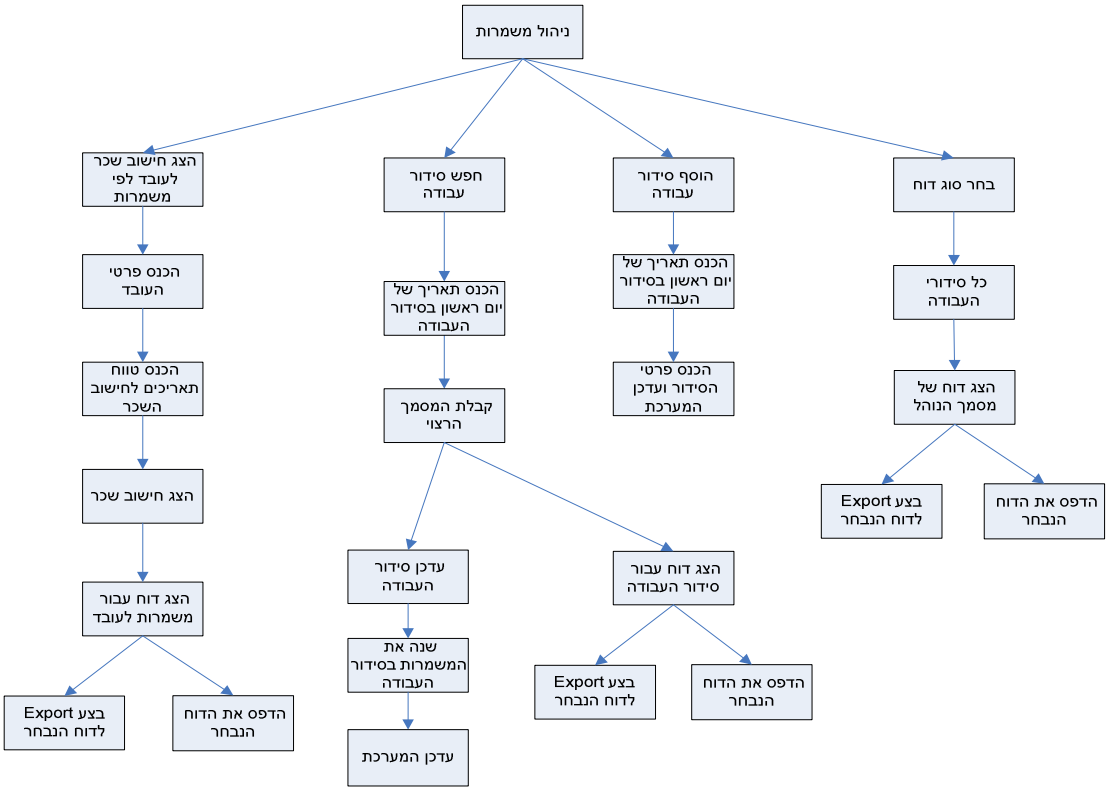
- הוספת רוטינת טיפול חדשה ושמירתו במאגר הנתונים
- הגדרת מסמך ותיוק רוטינת טיפול מתאימה
- שליפת נוהל עבודה קיים ועדכונו במערכת
- ביצוע חיפוש דינמי לפי פרמטרי נוהל העבודה
- הצגה גראפית של התפלגות נהלי העבודה לפי סוגיהם
- תיוק נוהל ספציפי כמענה לפרמטרי אירוע חריג
- יצירת רוטינת טיפול דינאמית למעקב בעת טיפול באירוע
- שליחת מייל לעובדים הרלוונטיים עם פרטי נוהל העבודה
- הפקת דו"חות על נהלי העבודה השונים במערכת

2.2.4 ניהול משמרות:

Presentation Layer:

איור 8 – ממשק גראפי של מודול ניהול הידע

Business Logic Layer: ניהול משמרות



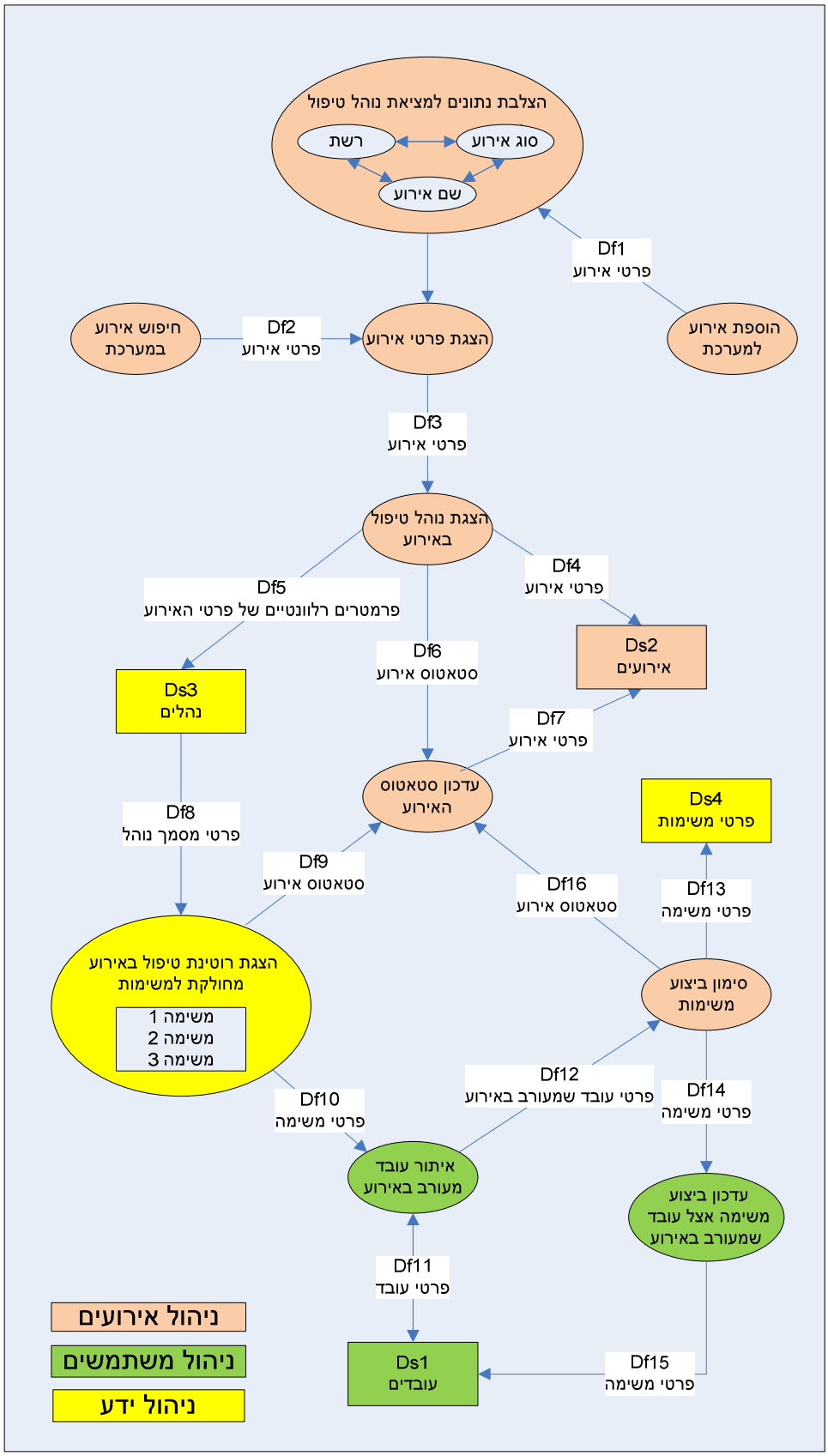
איור 9 – פונקציונאליות בשכבת הלוגיקה של מודול ניהול המשמרות

מודול ניהול המשמרות (איור 8 + איור 9 מאפשר לבצע את הפעולות הבאות:

- הוספת סידור עבודה חדש
- שליפת סידור עבודה קיים ועדכונו במערכת
- חישוב שכר לכל עובדי המוקד במתן טווח תאריכים שבו העובד רוצה לחשב את משכורתו
- הפקת דו"חות על סידורי העובדה הקיימים במערכת

התממשקות בין מודולי המערכת

ההתממשקות הפנימית בין המודולים באה לידי ביטוי באלגוריתם להצגת ותיעוד נוהל טיפול באירוע באופן הבא:



איור 10 – פונקציונאליות האלגוריתם להצגת ותיעוד נוהל טיפול באירוע

<u>תיאור מילולי של האלגוריתם להצגת ותיעוד נוהל טיפול באירוע</u>

1. הוסף אירוע חדש למערכת
2. לצורך מציאת נוהל טיפול מתאים, המערכת תבצע הצלבת נתונים בעזרת הפרמטרים הבאים:
   - 2.1 שם האירוע
   - 2.2 רשת שבה אירוע האירוע (Domain)
   - 2.3 סוג האירוע
3. הצג פרטי האירוע של האירוע הנבחר לצורך טיפול
4. הצג נוהל טיפול האירוע
5. המערכת תציג נוהל מתאים לאירוע המבוקש
6. רוטינת הטיפול תחולק לתתי משימות שאותן יש לבצע לפי הסדר שהגדירה המערכת
7. לצורך ביצוע המשימות המערכת תאתר את פרטי העובדים שמעורבים בנוהל הטיפול באירוע
8. הבקר יסמן את המשימות שביצע ויעדכן את המערכת
   - 8.1 אם (כל המשימות בוצעו)
     - 8.1.1.1 המערכת תעדכן את סטאטוס האירוע כסגור
     - 8.1.1.2 המערכת תעדכן את פרטי המשימה שבוצעו
     - 8.1.1.3 המערכת תעדכן ביצוע משימות אצל העובדים הרלוונטיים
   - 8.2 אחרת
     - 8.2.1.1 המערכת תפתח חלון להוספת הערות אודות הטיפול באירוע בו הבקר יציין איזה משימות לא בוצעו מתוך נוהל הביצוע.
     - 8.2.1.2 המערכת תעדכן את סטאטוס האירוע כפתוח לצורך השלמת הטיפול
     - 8.2.1.3 המערכת תעדכן את פרטי המשימה שבוצעו
     - 8.2.1.4 המערכת תעדכן ביצוע משימות אצל העובדים הרלוונטיים
9. הקש סיים נוהל
10. המערכת תחזור למסך הראשי שיציג את האירועים הפתוחים של המערכת לפי רמת חומרתם.

<u>הערה</u>: נוהלי הטיפול יתעדכנו בדו"ח חפיפה בין המשמרות על מנת לעדכן את הבקרים בעבודה השוטפת

3. Detailed Description Of Component

3.1. תפעול בסיס נתונים (DBManipulate)

3.1.1. מחלקה
3.1.2. מכילה את הפונקציות שאחראיות על הצגת טבלאות ובדיקת נוכחות ערך מסוים באחד הטבלאות בבסיס הנתונים (כל מודולי המערכת עובדים עם קומפוננט זה).
3.1.3. בחלק מן המתודות המופעלות על הקומפוננט פרמטר הוא שאילתת SQL והפלט הוא DataTable המכיל את הטבלה עם הפרטים שהתבקשו על פי השאילתה, מתודות אחרות אחראיות על פתיחת קישור לבסיס הנתונים וסגירתו. להלן חלק ממתודות Public שמופעלות ע"י הקומפוננט:

3.1.3.1. מתודה שמקבלת מחרוזת המכילה שאילתת SQL (משפט select) ומחזירה נתוני טבלה באמצעות DataTable

```
public DataTable Query(string query)
```

3.1.3.2. מתודה שמקבלת מחרוזת המכילה שאילתת SQL (משפט select) ומחזירה נתוני טבלה באמצעות DataSet

```
public DataSet SetQuery(string query)
```

3.1.3.3. מתודה שמקבלת רשימת מחרוזות המכילות שאילתות SQL (משפטי insert update או משפטי delete) ומחזירה ערך בוליאני המסמן הצלחת הפעולה. כל השאילתות עטופות בטרנזקציה

```
public string dbMultiAction(string[] sqlQueryList)
```

3.1.3.4. מתודה שמקבלת מחרוזת המכילה שאילתת SQL (משפט insert update או משפט delete) ומחזירה ערך בוליאני המסמן הצלחת הפעולה

```
public bool NonQuery(String DBstring)
```

3.1.4.4. מתודה האחראית על סגירת קישור לבסיס הנתונים

```
public void closeConnection()
```

3.1.1. N/A
3.1.2. N/A
3.1.3. תפקיד הקומפוננט הוא לקשר בין בסיס הנתונים לשכבת הלוגיקה, לצורך העבודה השוטפת של ביצוע פעולות על בסיס הנתונים.
3.1.4. הנתונים השייכים לקומפוננט הן: SqlConnection האחראי על הקמת קישור לבסיס הנתונים ומשתנה בוליאני isConnected הבודק האם קיים קישור לבסיס הנתונים.

3.2. שירות שכבת הלוגיקה (BLService)

3.2.1. מחלקה
3.2.2. מכילה את האובייקטים שאחראים על קישור לבסיס הנתונים.
3.2.3. בחלק מן המתודות המופעלות על הקומפוננט פרמטר הוא שאילתת SQL והפלט הוא DataTable המכיל את הטבלה עם הפרטים שהתבקשו על פי בשאילתה, במתודות אחרות הקלט הוא שדה המפתח הראשי שהוא של טבלה מסוימת והפלט הוא משתנה בוליאני המציין נוכחותו של משתנה זה בטבלה. להלן חלק ממתודות Public שמופעלות ע"י הקומפוננט:

3.9.3.1. מתודה שמקבלת פרמטר שם עמודה, ערך מזהה מתאים וטבלה מבסיס הנתונים ומחזירה ערך בוליאני המציין האם הנתון קיים בטבלה

```
public bool ChkIdInTable(string colName, string id,
string table)
```

3.9.3.2. מתודה שמקבלת ת"ז עובד וסיסמתו ובודקת האם הסיסמא מתאימה לשם המשתמש, מתודה זאת משמשת לבדיקת Log In למערכת

```
public bool ChkLogin(string id, string password)
```

3.9.3.3. מתודה שמקבלת פרמטר שם עמודה, ערך מזהה מתאים וטבלה מבסיס הנתונים ומוחקת את רשומת הערך הנתון מבסיס הנתונים

```
public void DeleteFromTable(string colName, string id,
string table)
```

3.9.3.4. מתודה שמקבלת כמחרוזת שם טבלה בבסיס הנתונים ומחזירה נתוני כל טבלה באמצעות DataTable

```
public DataTable ShowTable(string table)
```

3.9.3.5. מתודה שמקבלת פרמטר שם עמודה, ערך מזהה מתאים וטבלה מבסיס הנתונים ומחזירה נתוני הערך המבוקש באמצעות DataTable

```
public DataTable FindInTable(string colName, string id,
string table)
```

3.9.3.6. מתודה לבדיקת נוכחות פרוצדורה בטבלה

```
public bool ChkAttachedProcInTable(string colName, string
id, string table)
```

3.9.3.7. מתודה לבדיקת סטאטוס פרוצדורה

```
public string ChkProcStatusInTable(string colName, string
id, string table)
```

3.9.3.8. מתודה להצגת טבלה ממוינת לפי עמודה

```
public DataTable ShowTableSortByColName(string table,
string colName)
```

3.9.3.9. מתודה להצגת טבלה ממוינת לפי תקופה מתבקשת

```
public DataTable
ShowTableSortByColNameLastPeriod(string table, string
colName, string currentDate, string lastDate)
```

3.9.3.10. מתודה להצגת טבלה ממוינת לפי סטאטוס

```
public DataTable ShowTableSortByColNameStatus(string
table, string colName)
```

3.9.3.11. מתודה למציאת שם משתמש לפי ת"ז עובד

```
public string FindWorkerNameByID(string id)
```

3.9.3.12. מתודה למציאת תפקיד משתמש לפי ת"ז עובד

```
public string FindWorkerRoleByID(string id)
```

3.9.3.13. מתודה למציאת שכר בסיס לפי שם עובד

```
public int FindWorkerSalaryByName(string WorkerName)
public DataTable FindInTableSortByColName(string colName,
string id, string table, string SortCol)
```

3.9.3.14. מתודה למציאת נתונים בהינתן טווח תאריכים

```
public DataTable FindBetweenDates(string colName, string
from , string to, string table, string sortCol)
```

3.9.3.15. מתודה למציאת משמרות עובד לפי שם עובד

```
public DataTable FindShiftsByWorkerName(string
workerName, string from, string to)
```

3.9.3.16. מתודה להצגת נתוני טבלה נתונה

```
public string[] GetCollumnDataInTable(string colName,
string tableName)
```

3.9.3.17. מתודה להצגת נתוני טבלה נתונה בהינתן עמודה ריקה

```
public string[] GetCollumnDataInTableWhereNull(string
colName, string nullColName, string tableName )
```

3.9.3.18. מתודה להצגת נתוני טבלה נתונה בהינתן תנאי

```
public string[]
GetCollumnDataInTableWhereCondition(string colName,
string conditionCol , string conditionValue, string
tableName)
```

3.9.3.19. מתודה לביצוע סינון בחיפוש דינאמי

```
public object getResult(string filter, string
tableName, string colName
```

3.9.3.20. מתודה לביצוע סינון בחיפוש דינאמי לתוך Data Table

```
public DataTable getResultTable(string filter, string
tableName, string colName)
```

3.9.3.21. מתודה לביצוע סינון בחיפוש דינאמי לפי תאריכים לתוך Data Table

```
public DataTable getResultTableByDate(string filter,
string tableName, string colName)
```

3.9.3.22. מתודה לביצוע סינון בחיפוש דינאמי ממוין לפי תאריכים

```
public object getResultByDate(string filter, string
tableName, string colName)
```

3.9.3.23. מתודה להמרת תאריך אמריקאי לישראלי

```
public string ConvertDate(string date)
```

3.9.3.24. מתודה להצפנת סיסמא

```
public string Encrypt(string password)
```

3.9.3.25. מתודה לבדיקה התאמה בין סיסמאות

```
public bool Match(string encPSW, string basePSW)
```

3.9.3.26. מתודה להצפנת סיסמא למערך ביטים

```
private byte[] encryptPassword(string password)
```

3.9.3.27. מתודה למציאת אימייל לפי מספר עובד

```
public string FindEmailByWorkerNumber(string num)
```

3.9.3.28. מתודה לשליחת מייל ליעד בודד

```
public string SendSingleEmail(string mailUser, string
mailPassword, string to, string from, string subject,
string body)
```

3.9.3.29. מתודה לשליחת מייל להרבה יעדים

```
public string SendMultiEmail(string mailUser, string
mailPassword, string[] to, string from, string
subject, string body)
```

3.9.4. N/A

3.9.5. N/A

3.9.6. תפקיד הקומפוננט להעביר נתונים מתוך בסיס הנתונים לשכבת הלוגיקה.

3.9.7. הנתונים השייכים לקומפוננט הן: DBManipulate האחראי על הקמת קישור לבסיס הנתונים וביצוע פעולות על בסיס הנתונים ומשתנה סטאטי blService האחראי לביצוע פעולות הקומפוננט.

3.3. אירוע (Event)

3.3.1. מחלקה
3.3.2. מכילה את פרטי האירוע לצורך העבודה השוטפת של המערכת.
3.3.3. בחלק מן המתודות המופעלות על האירוע פרמטרי הקלט הם פרטי האירוע עצמו עצמה והפלט הוא שדה בוליאני שמסמן הצלחה או כשלון, במתודות אחרות הקלט הוא שדה המפתח של האירוע והפלט הוא פרטי האירוע עצמו. להלן חלק ממתודות Public שמופעלות ע"י הקומפוננט:

3.3.3.1. מתודה שמקבלת פרמטר שם עמודה וערך מזהה מתאים ומחזירה נתוני האירוע המבוקש

```
public Event FindEvent(string colomnName, string id)
```

3.3.3.2. מתודה שמקבלת אובייקט אירוע ומוספה את נתוני האירוע לטבלת האירועים בבסיס הנתונים

```
public void AddEvent(Event ev)
```

3.3.3.3. מתודה שמקבלת אובייקט אירוע ומעדכנת את נתוני האירוע בטבלת האירועים בבסיס הנתונים

```
public void UpdateEvent(Event ev)
```

3.3.3.4. מתודה שמקבלת אובייקט אירוע ומוחקת את נתוני האירוע מטבלת האירועים בבסיס הנתונים

```
public void DeleteEvent(Event ev)
```

3.3.3.5. מתודה שמוצאת אירועים בין 2 תאריכים נתונים

```
public Event FindEventBetweenDates(string from,
string to)
```

3.3.3.6. מתודה שמעדכנת רוטינת טיפול באירוע

```
public void updateEventTreatmentStatus(Event ev,
TreatProcedure treatproc, TreatmentItem[] tretItemArray)
```

3.3.4. N/A
3.3.5. N/A
3.3.6. כל הפעולות הנעשות על קומפוננט האירוע מתבצעות במסגרת מודול ניהול האירועים שהפונקציונאליות שלו מתוארת במפורט בסעיף 2.2.1 בנספח זה.
3.3.7. הנתונים השייכים לקומפוננט האירוע הן פרטי האירוע הבאים: קוד אירוע, שם אירוע, סוג אירוע, תאריך התרחשות האירוע, זמן התרחשות, שם המחשב שבו אירע האירוע, שם משתמש שביצע את האירוע, מקור האירוע וקוד נוהל טיפול באירוע.

3.4. משתמש (User/Worker)

3.4.1. מחלקה
3.4.2. מכילה את פרטי המשתמש לצורך העבודה השוטפת של המערכת.
3.4.3. בחלק מן המתודות המופעלות על המשתמש פרמטרי הקלט הם פרטי המשתמש עצמו עצמה והפלט הוא שדה בוליאני שמסמן הצלחה או כשלון, במתודות אחרות הקלט הוא שדה המפתח של המשתמש והפלט הוא פרטי המשתמש עצמו. להלן חלק ממתודות Public שמופעלות ע"י הקומפוננט:
3.4.3.1. מתודה שמקבלת אובייקט עובד ומוספה את נתוני העובד לטבלת העובדים בבסיס הנתונים

```
public void AddWorker(Worker w)
```

3.4.3.2. מתודה שמקבלת פרמטר שם עמודה וערך מזהה מתאים ומחזירה נתוני המשתמש המבוקש

```
public Worker FindWorker(string columnName, string id)
```

3.4.3.3. מתודה שמקבלת אובייקט עובד ומעדכנת את נתוני העובד בטבלת העובדים בבסיס הנתונים

```
public void UpdateWorker(Worker w)
```

3.4.3.4. מתודה שמקבלת אובייקט עובד ומוחקת את נתוני העובד מטבלת העובדים בבסיס הנתונים

```
public void DeleteWorker(Worker worker)
```

3.4.4. N/A

3.4.5. N/A

3.4.6. כל הפעולות הנעשות על קומפוננט המשתמש מתבצעות במסגרת מודול ניהול המשתמשים שהפונקציונאליות שלו מתוארת במפורט בסעיף 2.2.2 בנספח זה.

3.4.7. הנתונים השייכים לקומפוננט המשתמש הם פרטי העובד הבאים: ת"ז עובד, מספר עובד, שם משפחה, שם פרטי, שם משתמש, כתובת, טלפון, אימייל, תאריך לידה, תאריך תחילת עבודה, שכר בסיס, סטאטוס במערכת (פעיל או לא פעיל),קוד תפקיד וקוד מחלקה.

3.5. מסמך נוהל (Document)

3.5.1. מחלקה

3.5.2. מכילה את פרטי מסמך הנוהל לצורך העבודה השוטפת של המערכת.

3.5.3. בחלק מן המתודות המופעלות על קומפוננט מסמך הנוהל פרמטרי הקלט הם פרטי הנוהל עצמו עצמה והפלט הוא שדה בוליאני שמסמן הצלחה או כשלון, במתודות אחרות הקלט הוא שדה המפתח של מסמך הנוהל והפלט הוא פרטי הנוהל עצמו. להלן חלק ממתודות Public שמופעלות ע"י הקומפוננט:

3.5.3.1. מתודה שמקבלת אובייקט מסמך נוהל ומוחקת את נתוני מסמך הנוהל מטבלת המסמכים בבסיס הנתונים

```
public void DeleteDocument(Document doc)
```

3.5.3.2. מתודה שמקבלת אובייקט מסמך נוהל ומעדכנת את נתוני מסמך הנוהל בטבלת המסמכים בבסיס הנתונים

```
public void UpdateDocument(Document doc)
```

3.5.3.3. מתודה שמקבלת פרמטר שם עמודה וערך מזהה מתאים ומחזירה נתוני הנוהל המבוקש

```
public Document FindDocument(string ColumnName,string id)
```

3.5.3.4. מתודה שמקבלת אובייקט מסמך נוהל ומוספה את נתוני מסמך הנוהל לטבלת המסמכים בבסיס הנתונים

```
public void AddDocument(Document doc)
```

3.5.4. N/A

3.5.5. N/A

3.5.6. כל הפעולות הנעשות על קומפוננט האירוע מתבצעות במסגרת מודול ניהול הידע שהפונקציונאליות שלו מתוארת במפורט בסעיף 2.2.3 בנספח זה.

3.5.7. הנתונים השייכים לקומפוננט מסמך הנוהל הן פרטי הנוהל הבאים: קוד מסמך נוהל, שם המסמך, תיאור המסמך ותכולת המסמך.

3.6. נוהל טיפול באירוע (Treatment Procedure)

3.6.1. מחלקה

3.6.2. מכילה את פרטי נוהל טיפול באירוע.

3.6.3. בחלק מן המתודות המופעלות על קומפוננט נוהל טיפול באירוע פרמטר הקלט הוא קוד נוהל טיפול עצמו והפלט הם פרטי האירוע, במתודות אחרות הקלט הוא שדה המפתח הזר שהוא ת"ז עובד שטיפל באירוע והפלט הוא פרטי נהלי הטיפול שבהם טיפל.

3.6.3.1. מתודה להוספת פריטי רוטינת טיפול טיפול

```
public void AddTreatItems(TreatProcedure tp,
string[] tpItems, int i)
```

3.6.3.2. מתודה לעדכון פרטי רוטינת טיפול

```
public void UpdateTreatItems(TreatProcedure tp, string[]
tpItems, int i)
```

3.6.3.3. מתודה להוספת פרטי טיפול לרוטינת טיפול

```
public void AddProcItems(TreatProcedure tp, string[]
temp)
```

3.6.3.4. מתודה למציאת פרטי רוטינת טיפול

```
public string[] FindTreatItems(string ColumnName,
string id)
```

3.6.3.5. מתודה להוספת רוטינת טיפול

```
public void AddTreatProcudure(TreatProcedure
treatproc)
```

3.6.3.6. מתודה להוספת רוטינת טיפול עטופה בטרנזקציה

```
public void AddTreatProcudure(TreatProcedure
treatproc, TreatmentItem[] tretItemArray, string[]
treatItemsID)
```

3.6.3.7. מתודה לעדכון שם פרוצדורת טיפול

```
public void UpdateTreatName(TreatProcedure tp)
```

3.6.3.8.

3.6.4. N/A

3.6.5. N/A

3.6.6. תפקיד קומפוננט נוהל הטיפול מצד אחד הוא להציג נוהל טיפול לאירוע שנרשם במערכת ומצד שני לנהל מעקב אחרי העובדים שביצעו נהלי טיפול שונים לאירועים שנרשמו במערכת.

3.6.7. הנתונים השייכים לקומפוננט נוהל הטיפול הן פרטי נוהל הטיפול הבאים: קוד נוהל טיפול באירוע, שם נוהל טיפול באירוע, תיאור הטיפול, סטאטוס הטיפול ות"ז מטפל האירוע.

3.7. פריט מטופל (Treatment Item)

3.7.1. מחלקה

3.7.2. מכילה פריט טיפול באירוע.

3.7.3. המתודות המופעלות על קומפוננט פריט הטיפול מקבלות פרמטר קלט של קוד פריט הטיפול ומחזירות את פרטי פריט הטיפול שהם תיאור הפריט, תיאור הטיפול וסטאטוס הטיפול.

3.7.3.1. מתודה למציאת פריט טיפול

```
public TreatmentItem FindItem(string ColumnName,
string id)
```

3.7.3.2. מתודה לעדכון תיאור הטיפול

```
public void UpdateDescriptionItem(TreatmentItem[]
tretItemArray)
```

3.7.3.3. מתודה לעדכון עובד המעורב בפריט הטיפול

```
public void UpdateWorkerItem(TreatmentItem tretItem)
```

3.7.3.4. מתודה לעדכון סטאטוס הטיפול

```
public void UpdateStatusItem(TreatmentItem tretItem)
```

3.7.3.5. מתודה להוספת פרטי טיפול

```
public void AddTreatItems(TreatmentItem treatItem)
```

3.7.4. N/A

3.7.5. N/A

3.7.6. תפקיד קומפוננט פריט הטיפול הוא להרכיב ממספר פריטי טיפול שונים נוהל טיפול מורכב האחראי לטפל באירוע שנרשם במערכת.

3.7.7. הנתונים השייכים לקומפוננט פריט הטיפול הן פרטי פריט הטיפול הבאים: קוד פריט מטופל, תיאור הפריט, תיאור הטיפול וסטאטוס הטיפול.

3.8. משמרת (Shift Schedule)

3.8.1. מחלקה
3.8.2. מכילה נתונים עבור משמרות הנרשמות במערכת.
3.8.3. המתודות העיקרית של קומפוננט האירוע הם הוספת סידור עבודה חדש, עדכון סידור עבודה קיים, מציאת סידור עבודה נוכחי וחישוב שכר לעובדים על פי המשמרות בהם עבדו
3.8.3.1. מתודה להוספת סידור עבודה

```
public void AddShift(ShiftSchedule [] shiftArray)
```

3.8.3.2. מתודה לעדכון סידור עבודה

```
public void UpdateShift(ShiftSchedule [] shiftArray)
```

3.8.3.3. מתודה למציאת סידור עבודה עדכני

```
public ShiftSchedule FindShift(string ColumnName,
string id)
```

3.8.3.4.
3.8.4. N/A
3.8.5. N/A
3.8.6. קומפוננט המשמרת אחראית על ניהול שותף של סידור העבודה במוקד כולל הפקד חישובי שכר לעובדים עבור טווח תאריכים נתון
3.8.7. הנתונים השייכים לקומפוננט משמרת הן פרטי המשמרת הבאים: קוד משמרת, תאירך משמרת, בקר1 ובקר 2 שעובדים במשמרת.

3.9. אלגוריתם לתיוק נוהל טיפול (DecideTreatAlgorithm.cs)

3.9.1. מחלקה
3.9.2. תפקידה למצוא נוהל טיפול מתאים בהתאם לנתוני האירוע.
3.9.3. המתודה העיקרית של הקומפוננט מוצאת רוטינת טיפול מתאימה בהתאם לפרמטרי האירוע הנתון
3.9.3.1. מתודה שמוצאת רוטינת טיפול

```
public string GetTreatProcedure(string eventTypeID,
string eventName, string domain)
```

3.9.3.2. מתודה שמוסיפה רוטינת טיפול חדשה

```
public void AddTreatAlgorithm(TreatProcedure
treatproc, string treatProcedureID, string
eventTypeID, string eventName, string domain)
```

3.9.4. N/A
3.9.5. N/A
3.9.6. קומפוננט האלגוריתם אחראי להחזיר רוטינת טיפול בהתאם לפרטי האירוע הנרשם במערכת
3.9.7. N/A

4.1. חלון כניסה למערכת (Log In)

4.1.1. חלון כניסה למערכת.

4.1.2.

איור 11 – חלון כניסה למערכת

4.1.3. כניסה למערכת באמצעות הכנסת שם משתמש וסיסמא.

4.2. חלון מערכת מרכזי (Start Form)

4.2.1. חלון מערכת מרכזי המכיל קישור למנהלים השונים.

4.2.2.

איור 12 – חלון מערכת מרכזי

4.2.3. חלון זה יכיל את מאזן האירועים האחרונים שרשמו במערכת והתפלגות האירועים לפי סטאטוס וחומרת האירועים. בנוסף חלון זה מאפשר גישה למודולי ניהול האירועים, ניהול המשתמשים וניהול הידע וניהול משמרות.

4.3. חלון ניהול אירועים (Event Manager)

4.3.1. חלון מנהל אירועים המכיל פעולות שונות על קומפוננט האירוע.

4.3.2.

איור 13 – חלון מנהל אירועים

4.3.3. צפייה באירועים שנרשמו במערכת, חיפוש אירועים לפי פרמטרים,צפיית פרטי אירוע ספציפי, עדכון פרטי אירוע, מחיקת אירוע מהמערכת והפקת דוחות.

4.4. חלון הצגת פרטי אירוע (Show/Update Event)

4.4.1. חלון הצגת פרטי האירוע בהתאם לסוג האירוע.

4.4.2.

איור 14 – חלון הצגת פרטי אירוע לדיווח

4.4.3. צפייה בפרטי האירוע בהתאם לסוגו, אפשרות לראות אירוע קודם או אירוע הבא, עדכון, מחיקה, הצגת נוהל טיפול באירוע והפקת דוחות.

4.5. חלון הצגת נוהל טיפול באירוע (Handle Event)

4.5.1. חלון הצגת נוהל טיפול באירוע המאפשר לעדכן סטאטוס אירוע בסיום הטיפול.

4.5.2.

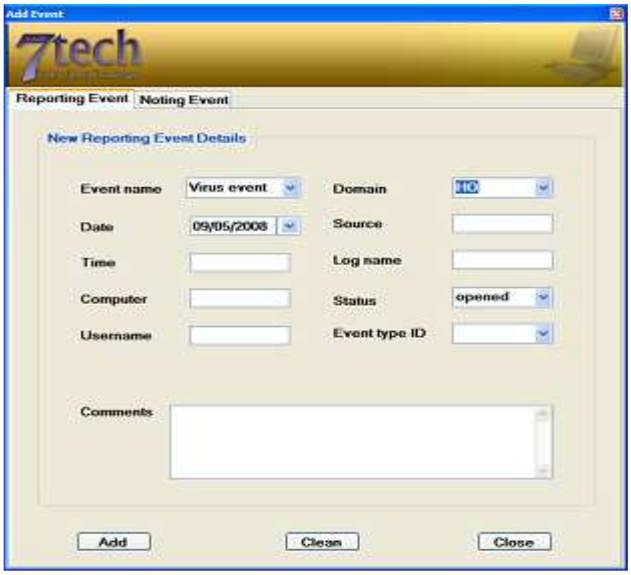
איור 15 – חלון הצגת נוהל טיפול באירוע

4.5.3. עדכון ביצוע פרטי הנוהל, כאשר לאחר ביצוע העדכון המערכת תעדכן את סטאטוס האירוע.

4.6. חלון הוספת אירוע (Add Event)

4.6.1. חלון הוספת אירוע חדש למערכת.

4.6.2.



איור 16 – חלון הוספת אירוע

4.6.3. הכנסת פרטי האירוע (בהתאם לסוגו) ועדכון רישום האירוע במערכת.

4.7. חלון ניהול משתמשים (User Manager)

4.7.1. חלון מנהל משתמשים המכיל פעולות שונות על קומפוננט המשתמש
4.7.2.

איור 18 – חלון ניהול משתמשים

4.7.3. צפייה במשתמשים הקיימים במערכת, חיפוש משתמשים לפי פרמטרים,צפיית פרטי משתמש ספציפי, עדכון פרטי משתמש, מחיקת משתמש קיים מהמערכת והפקת דוחות.

4.8. חלון הצגת פרטי משתמש (Show/Update User)

4.8.1. חלון הצגת פרטי המשתמש.
4.8.2.

איור 19 – חלון הצגת פרטי משתמש

4.8.3. צפייה בפרטי המשתמש, אפשרות לראות משתמש קודם או משתמש הבא, עדכון, מחיקת משתמש קיים.

4.9. חלון הוספת משתמש (Add User)

4.9.1. חלון הוספת משתמש חדש למערכת.

4.9.2.

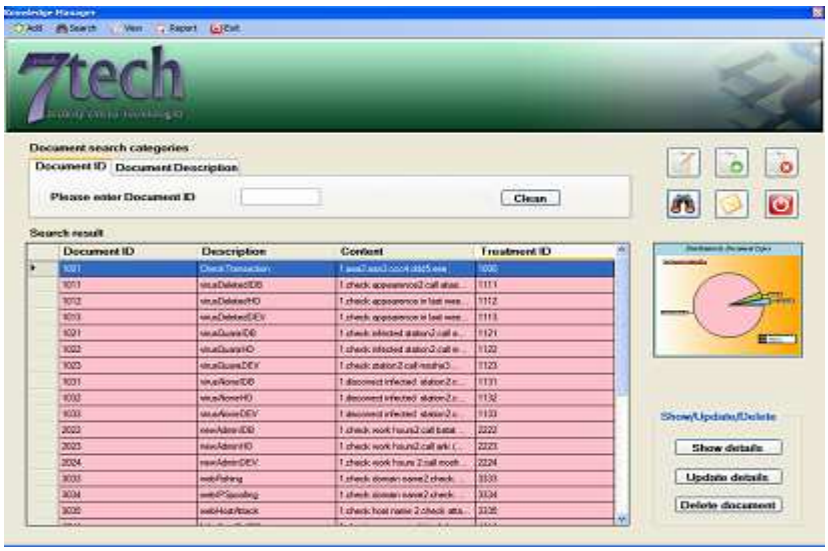
איור 20 – חלון הוספת משתמש

4.9.3. הכנסת פרטי המשתמש ועדכון רישום המשתמש במערכת.

4.10. חלון ניהול ידע (Knowledge Manager)

4.10.1. חלון מנהל ידע המכיל פעולות שונות על קומפוננט מסמך נוהל.

4.10.2.



איור 21 – חלון ניהול ידע

4.10.3. צפייה במסמכי נהלים שנרשמו במערכת, חיפוש מסמכים לפי פרמטרים שונים,צפיית פרטי נוהל ספציפי, עדכון פרטי נוהל, מחיקת נוהל קיים מהמערכת והפקת דוחות.

4.11. חלון הצגת מסמך נוהל (Show/Update Document)

4.11.1. חלון הצגת פרטי מסמך נוהל.

4.11.2.

איור 22 – חלון הצגת מסמך נוהל

4.11.3. צפייה בפרטי נוהל, אפשרות לראות נוהל קודם או נוהל הבא, עדכון, מחיקת משתמש קיים והפקת דוחות.

4.12. חלון הוספת מסמך נוהל חדש (Add Document)

4.12.1. חלון הוספת מסמך נוהל חדש למערכת.

4.12.2.

איור 23 – חלון הוספת מסמך נוהל חדש

4.12.3. הכנסת פרטי מסמך הנוהל ועדכון רישום המסמך במערכת.

4.13. חלון מנהל משמרות (Shift Manager)

4.13.1. חלון מנהל משמרות.

4.13.2.

איור 24 – חלון הוספת מסמך נוהל חדש

4.13.3. צפייה בסידור עבודה נוכחי עם אפשרות להכניס סידור עבודה חדש, לעדכן סידור עבודה קיים, לחשב שכר לעובד בהתאם למשמרותיו והפקת דוחות רלוונטיים.

4.14. חלון תיוק פרוצדורת טיפול (Attack procedure)

4.14.1. חלון תיוק פרוצדורה חדשה.

4.14.2.

איור 25 – חלון הוספת מסמך נוהל חדש

4.14.3. הכנסת פרטי פרוצדורה מתאימה לפי פרמטרי האירוע.

4.15. חלון להצגת דוחות (Report Form)

4.15.1. חלון להצגת דוחות מערכת.

4.15.2.

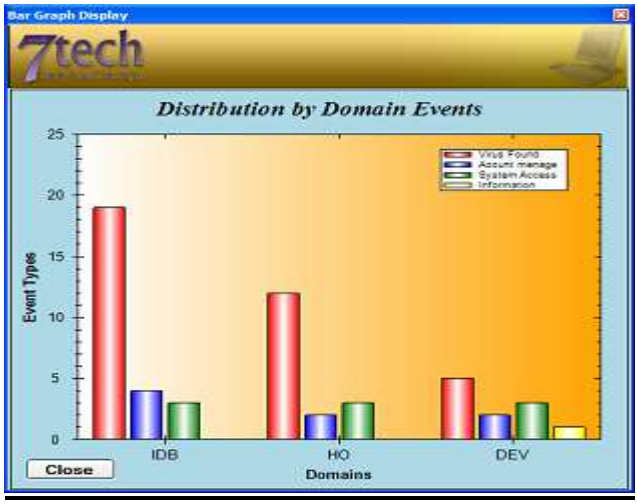
איור 26 – חלון הוספת מסמך נוהל חדש

4.15.3. הכנסת פרטי דוח להצגה ויזואלית.

4.16. חלון להצגת גרפים (Show Graph)

4.16.1. חלון להצגת גרפים.

4.16.2.



איור 27 – חלון הוספת מסמך נוהל חדש

4.16.3. הכנסת פרטי הגרף הרצוי להצגה ויזואלית.

4.16.4.

4.17. חלון איפוס סיסמת משתמש (Reset Password)

4.17.1. חלון איפוס סיסמת משתמש.

.4.17.2

איור 28 – חלון הוספת מסמך נוהל חדש

4.17.3. הכנסת ת"ז משתמש וסיסמא חדשה.

4.18. חלון קטגוריות להפקת דוחות (Report Categories)

4.18.1. חלון קטגוריות להפקת דוחות.

.4.18.2

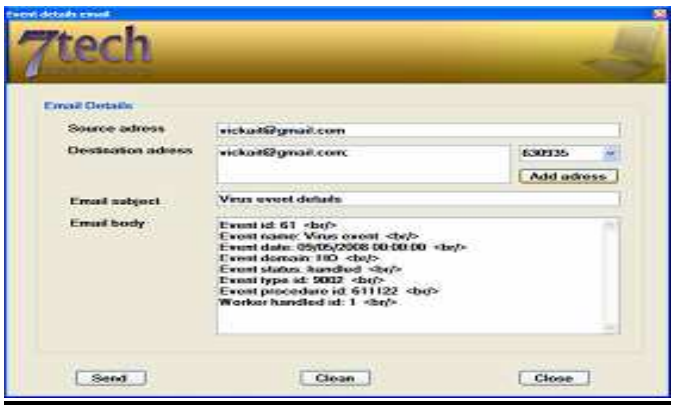
איור 29 – חלון הוספת מסמך נוהל חדש

4.18.3. הכנסת הפרטים הרלוונטיים להפקת הדוח.

4.19. חלון שליחת אימייל (Email Form)

4.19.1. חלון שליחת מייל.

.4.19.2



איור 30 – חלון שליחת מייל

4.19.3. החלון מכיל את כל השדות הרלוונטיים לשליחת מייל.

5. N/A - Additional Material

נספח ג - Software Test Documentation (STD)

1. Introduction

מסמך תכנון ועיצוב הבדיקות מוגדר לפי המודולים המרכזיים של המערכת כלומר לכל מודול מרכזי תהיה סדרת בדיקות משלו וכמובן בסופו של דבר תיערך סדרת בדיקות למערכת כולה.

2. Test items

מה שייבדק בשלב הבדיקות הוא קוד המקור שבאמצעותו תפותח המערכת כאשר הבדיקה יחולקו לשלבים הבאים:

1. בדיקות יחידה (Unit Tests) - בשלב זה יהיו בדיקות יחידה של המודולים השונים. מטרת בדיקות אלה היא לוודא שכל יחידת תוכנה בפני עצמה מבצעת את אשר תוכננה לבצע (למשל: תקינות עבודתם של stored procedures בשכבת בסיס הנתונים).

2. בדיקות אינטגרציה של יחידות תוכנה (Integration Tests) - המטרה היא לבדוק "שרשרת" של יחידות תוכנה, כדי להבטיח שהן "מדברות" זו עם זו בצורה תקינה, שהיחידות מסונכרנות זו עם זו וכן שהנתונים או הפרמטרים עוברים ביניהן בצורה תקינה (למשל: תקינות הפעלתו של stored procedures מתוך שכבת הלוגיקה עם העברת פרמטרים מסוימים לקבלת תוצאה רצויה).

3. בדיקות מערכת (System Tests) - בשלב זה המטרה היא לבדוק את המערכת כולה בהיבט הפונקציונאלי והטכני. מאחר והבדיקות מתבססות על מסמכי הדרישות של הלקוח, ניתן דרכן לזהות אי - עמידה בדרישות הלקוח (למשל: הופעת טבלה של כל האירועים בלחיצה על כפתור "הצג כל האירועים").

3. Features to be tested

המאפיינים שיבדקו בשלב הבדיקות הם:

3.1. נכונות - המידה שבה ממלאת מערכת המידע את כל מפרט הדרישות, ועומדת בדרישות התקנים והסטנדרטים לפיתוח תוכנה.
מערכת המידע תבדוק שהקלט, העיבוד והפלט, מוגדרים בצורה מלאה, ברורה ומדויקת, ושהכמות והחומרה של שגיאות התוכנה שואפת לאפס.

3.2. אמינות - בדיקה שמערכת המידע תפעל ללא תקלות.

3.3. יעילות - בדיקה של דרישות מערכת המידע למשאבי חומרה, כגון: דרישות למהירות מעבד, דרישות לגודל זיכרון ודרישות לנפח אכסון.
ככל שדרישות מערכת המידע למשאבי חומרה נמוכות יותר, אזי המערכת יעילה יותר.

3.4. אבטחתיות – בדיקה של אבטחת הנתונים במערכת המידע. המערכת תתמוך בהגבלת גישה.

3.5. שימושיות - ככל שההפעלה והשימוש בתוכנה יהיו קלים יותר, ידידותיים יותר, וידרשו פחות מזמנם של המשתמשים, כך תגדל השימושיות של מערכת המידע, כלומר וידוא שממשק המשתמש ידידותי ללקוח.

3.6. בדיקתיות - בדיקה שמתבצעת במהלך התפעול השוטף ובודקת שהתוכנית מבצעת את המטלות שהוגדרו לה. הבדיקה תכלול איתור תקלות בתוכנה, והגורם להם.

4. Features not to be tested

4.1. ניידות - לא תתבצע בדיקה של הפעלת המערכת על פלטפורמות שונות, מכיוון שהמערכת תלויה בסביבה שבה פותחה ולא מיועדת לעבוד על פלטפורמות אחרות.

4.2. התממשקות - לא תתבצע בדיקה של המערכת שתבדוק האם אפשר ליצור ממשקים עם תוכנות אחרות מכיוון שהמערכת עובדת ב Stand Alone ולא אמורה להתממשק עם מערכות אחרות.

5. Environmental needs

לצורך ביצוע הבדיקות נצטרך מחשב בעל מעבד פנטיום 4 ובעל זיכרון 1024MB ומעלה, מערכת ההפעלה שתותקן על המחשב היא Windows XP Professional. העורך שבו נשתמש לפיתוח המערכת וביצוע הבדיקות הוא Microsoft Visual Studio® .NET 2005 עם בסיס נתונים Microsoft SQL Server 2005 (שמובנה בעורך).

הבדיקות יתבצעו על מחשב יחיד עם בסיס נתונים מקומי מכוון שהמערכת לא תתממשק למערכות אחרות הפועלות במקביל במסגרת מוקד אבטחת המידע של בנק דיסקונט.

6. Test Cases

6.1 Case1

6.1.1 מטרה
תקינות הקמת קישור לשכבת בסיס הנתונים דרך שכבת DAL.
6.1.2 קלט
הכנסת ת"ז עובד ולחיצה על כפתור "הצג".
6.1.3 פלט מצופה (קריטריון הצלחה/כשלון)
הצגת שם העובד בתיבת טקסט.
קריטריון לכישלון – שגיאת SQL.
קריטריון להצלחה – קבלת שם עובד בעל ת"ז שהכנסנו כקלט.
6.1.4 תהליכי בדיקה
6.1.4.1 מריצים את סביבת הפיתוח.
6.1.4.2 נכנסים למערכת.
6.1.4.3 מכינים את ת"ז העובד לתוך תיבת הטקסט.
6.1.4.4 לוחצים "הצג" .
6.1.4.5 מוודאים שאין הודעת שגיאה.

6.2 Case2

6.2.1 מטרה
בדיקת הוספת מסמך חדש למאגר המסמכים.
6.2.2 קלט
הכנסת פרטי המסמך ולחיצה על כפתור "הוסף".
6.2.3 פלט מצופה (קריטריון הצלחה/כשלון)
הצגת הודעה: "המסמך צורף בהצלחה"
קריטריון לכישלון – הודעת שגיאת SQL.
קריטריון להצלחה – קבלת : "המסמך צורף בהצלחה".
6.2.4 תהליכי בדיקה
6.2.4.1 מריצים את סביבת הפיתוח.
6.2.4.2 נכנסים למערכת.
6.2.4.3 לוחצים על כפתור "מנהל הידע"
6.2.4.4 לוחצים על כפתור "הוסף מסמך".
6.2.4.5 מכניסים את פרטי הנוהל ולוחצים "הוסף".
6.2.4.6 מוודאים שאין הודעת שגיאה.

6.3 Case3

6.3.1 מטרה
בדיקת הצגת כל האירועים הקיימים במערכת.
6.3.2 קלט
לחיצת כפתור "הצג כל האירועים".
6.3.3 פלט מצופה (קריטריון הצלחה/כשלון)
הצגת טבלת האירועים באמצעות DataGridView.
קריטריון לכישלון – הודעת שגיאת SQL.
קריטריון להצלחה – הטבלה מוצגת במלואה.

6.3.4. <u>תהליכי בדיקה</u>
6.3.4.1. מריצים את סביבת הפיתוח.
6.3.4.2. נכנסים למערכת.
6.3.4.3. לוחצים על כפתור "מנהל האירועים"
6.3.4.4. לוחצים על כפתור " הצג כל האירועים".
6.3.4.5. מוודאים שאין הודעת שגיאה.

6.4. <u>Case4</u>

6.4.1. <u>מטרה</u>
בדיקת הכנסת קלט תקין בעת ביצוע חיפוש עובד.
6.4.2. <u>קלט</u>
הכנסת ת"ז עובד ולחיצה על כפתור "חפש".
6.4.3. <u>פלט מצופה (קריטריון הצלחה/כשלון)</u>
הצגת פרטי העובד באמצעות DataGridView.
קריטריון לכישלון – רקע תיבת הטקסט לאחר סיום הכנסת הנתונים אדומה והודעת שגיאה (הצבע מראה בדיקת קלט דינאמית בהתאם לתבנית הרצויה, במקרה זה מותר להכניס רק ספרות).
קריטריון להצלחה – רקע תיבת הטקסט לאחר סיום הכנסת הנתונים ירוקה והצגת פרטי העובד.
6.4.4. <u>תהליכי בדיקה</u>
6.4.4.1. מריצים את סביבת הפיתוח.
6.4.4.2. נכנסים למערכת.
6.4.4.3. לוחצים על כפתור "מנהל המשתמשים".
6.4.4.4. מכניסים את ת"ז העובד בתיבה של פרטי החיפוש.
6.4.4.5. לוחצים על כפתור "חפש".
6.4.4.6. מוודאים שאין הודעת שגיאה.

6.5. <u>Case5</u>

6.5.1. <u>מטרה</u>
בדיקת קיום מסמך בבסיס הנתונים לפני מחיקתו.
6.5.2. <u>קלט</u>
סימון המסמך הרצוי ב DataGridView ולחיצה על כפתור "מחק".
6.5.3. <u>פלט מצופה (קריטריון הצלחה/כשלון)</u>
הודעת הצלחה: "המסמך נמחק בהצלחה".
קריטריון לכישלון – הודעת שגיאה: "המסמך לא קיים בבסיס הנתונים".
קריטריון להצלחה – הודעת הצלחה ומחיקת המסמך מטבלת המסמכים.
6.5.4. <u>תהליכי בדיקה</u>
6.5.4.1. מריצים את סביבת הפיתוח.
6.5.4.2. נכנסים למערכת.
6.5.4.3. לוחצים על כפתור "מנהל הידע"
6.5.4.4. לוחצים על כפתור "הצג כל המסמכים".
6.5.4.5. מסמנים את המסמך הרצוי.
6.5.4.6. לוחצים "מחק".
6.5.4.7. מוודאים שאין הודעת שגיאה.

6.6. Case6

6.6.1. מטרה
בדיקת הצגת נתונים האירוע בצורה תקינה.
6.6.2. קלט
לחיצה כפולה על הרשומה של האירוע הרצוי ב DataGridView.
6.6.3. פלט מצופה (קריטריון הצלחה/כשלון)
הצגת חלון עם פרטי האירוע.
קריטריון לכישלון – הודעת שגיאה.
קריטריון להצלחה – פרטי האירוע המופיעים החלון זהים לרשומה שנחלצה בטבלה.
6.6.4. תהליכי בדיקה
6.6.4.1. מריצים את סביבת הפיתוח.
6.6.4.2. נכנסים למערכת.
6.6.4.3. לוחצים על כפתור "מנהל האירועים"
6.6.4.4. לוחצים על כפתור " הצג כל האירועים".
6.6.4.5. לוחצים לחיצה כולה על האירוע הרצוי.
6.6.4.6. מוודאים שנפתח חלון עם פרטי האירוע.

6.7. Case7

6.7.1. מטרה
בדיקת תקינות כפתורי "קודם"/"הבא" בחלון פרטי האירוע.
6.7.2. קלט
לחיצה על כפתור "קודם"/"הבא" בחלון פרטי האירוע.
6.7.3. פלט מצופה (קריטריון הצלחה/כשלון)
הצגת שם האירוע הבא לפי ה DataGridView במנהל האירועים.
קריטריון לכישלון – הודעת שגיאה (חשוב לציין שברגע שמגיעים לתחילת או לסוף ה DataGridView, מקבלים הודעה מתאימה).
קריטריון להצלחה – קבלת פרטי האירוע הקודם/הבא.
6.7.4. תהליכי בדיקה
6.7.4.1. מריצים את סביבת הפיתוח.
6.7.4.2. נכנסים למערכת.
6.7.4.3. לוחצים על כפתור "מנהל האירועים".
6.7.4.4. לוחצים על כפתור "הצג כל האירועים".
6.7.4.5. לוחצים לחיצה כפולה על אחד הרשומות ומקבלים חלון פרטי האירוע.
6.7.4.6. לוחצים על כפתור "קודם"/"הבא".
6.7.4.7. מוודאים שקיבלנו פרטי אירוע נכונים.

6.8. Case8

6.8.1. מטרה
בדיקת קבלת נוהל נכון לטיפול באירוע נתון.
6.8.2. קלט
לחיצה על כפתור "הצג נוהל טיפול" במסך של פרטי האירוע.
6.8.3. פלט מצופה (קריטריון הצלחה/כשלון)
הצגת רוטינת נוהל טיפול באירוע.

קריטריון לכישלון – הודעת שגיאה: "לא קיים נוהל טיפול עבור סוג אירוע זה".
קריטריון להצלחה – הצגת פרטי נוהל לטיפול באירוע..

6.8.4. תהליכי בדיקה

6.8.4.1. מריצים את סביבת הפיתוח.
6.8.4.2. נכנסים למערכת.
6.8.4.3. לוחצים על כפתור מנהל האירועים.
6.8.4.4. לוחצים על כפתור " הצג כל האירועים".
6.8.4.5. לוחצים לחיצה כפולה על ה DataGridView ומקבלים פרטי אירוע.
6.8.4.6. לוחצים על כפתור "הצג נוהל טיפול".
6.8.4.7. מוודאים שאין הודעת שגיאה.

6.9. Case9

6.9.1. מטרה

בדיקת תיוק עובד לפריט טיפול.

6.9.2. קלט

לחיצה על כפתור "התערבות עובד" במסך של טיפול באירוע, הכנס מספר עובד ולחיצת "חפש".

6.9.3. פלט מצופה (קריטריון הצלחה/כשלון)

הצגת הודעה "העובד עודכן בהצלחה".
קריטריון לכישלון – הודעת שגיאה: "מספר העובד לא נמצא בבסיס הנתונים".
קריטריון להצלחה – הצגת הודעה "העובד עודכן בהצלחה" ולאחר מכן הצגת פרטי העובד.

6.9.4. תהליכי בדיקה

6.9.4.1. מריצים את סביבת הפיתוח.
6.9.4.2. נכנסים למערכת.
6.9.4.3. לוחצים על כפתור מנהל האירועים.
6.9.4.4. לוחצים על כפתור "הצג כל האירועים".
6.9.4.5. לוחצים לחיצה כפולה על ה DataGridView ומקבלים פרטי אירוע.
6.9.4.6. לוחצים על כפתור "הצג נוהל טיפול".
6.9.4.7. תוך כדי עדכון הטיפול באירוע לוחצים " Involve worker" ולאחר מכניסים מספר עובד ולוחצים "חפש".
6.9.4.8. מוודאים שאין הודעת שגיאה.

6.10. Case9

6.10.1. מטרה

בדיקת עדכון סטאטוס אירוע לאחר סגירת אירוע

6.10.2. קלט

סימון כל הפריטים בטיפול באירוע ולחיצת "Apply".

6.10.3. פלט מצופה (קריטריון הצלחה/כשלון)

הצגת הודעה "סטאטוס האירוע עודכן בהצלחה".
קריטריון לכישלון – הודעת שגיאה: "שגיאת SQL".
קריטריון להצלחה – הצגת הודעה: "סטאטוס האירוע עודכן בהצלחה" ובדיקה פעם נוספת בפרטי האירוע הסגור.

6.10.4. תהליכי בדיקה

6.10.4.1. מריצים את סביבת הפיתוח.
6.10.4.2. נכנסים למערכת.
6.10.4.3. לוחצים על כפתור מנהל האירועים.
6.10.4.4. לוחצים על כפתור "הצג כל האירועים".
6.10.4.5. לוחצים לחיצה כפולה על ה DataGridView ומקבלים פרטי אירוע.
6.10.4.6. לוחצים על כפתור "הצג נוהל טיפול".
6.10.4.7. מסמנים את כל פרטי הטיפול בנוהל המוצג ולוחצים "Apply".
6.10.4.8. מוודאים שאין הודעת שגיאה.

7. Schedule

שלב הבדיקות המרכזי שכולל בדיקות המערכת + תיקוני באגים נקבע בלוח הזמנים לתאריכים 07.02.08 - 20.02.08 שהם 10 ימי עבודה, כפי שניתן לראות בפירוט לוח הזמנים בנספח ג' - מסמך SPMP.

אך חשוב לציין כי חלק מהבדיקות היחידה והאינטגרציה נעשו בזמן תהליך פיתוח (שהם 50 ימי עבודה שנקבע בלוח הזמנים לתאריכים 26.09.07 - 04.12.07).

שלב הבדיקות (10 ימים) שאותו נבצע לאחר סיום בניית המערכת ולאחר וידוא שהמערכת עומדת בכל הדרישות מבחינת פונקציונאליות, ביצועים ואבטחת המידע יכלול את בדיקת המאפיינים שהוגדרו בפרק 3 בנספח זה והם:

1. נכונות
2. אמינות
3. יעילות
4. אבטחתיות
5. שימושיות
6. בדיקתיות

נספח ד - Software Test Report (STR)

1. Case1

1.1. מטרה
תקינות הקמת קישור לשכבת בסיס הנתונים דרך שכבת DAL.
1.2. קלט
הכנסת ת"ז עובד ולחיצה על כפתור "הצג".
1.3. פלט מצופה (קריטריון הצלחה/כשלון)
הצגת שם העובד בתיבת טקסט.
קריטריון לכישלון – שגיאת SQL.
קריטריון להצלחה – קבלת שם עובד בעל ת"ז שהכנסנו כקלט.
1.4. תהליכי בדיקה
1.4.1. מריצים את סביבת הפיתוח.
1.4.2. נכנסים למערכת.
1.4.3. מכינים את ת"ז העובד לתוך תיבת הטקסט.
1.4.4. לוחצים "הצג" .
1.4.5. מוודאים שאין הודעת שגיאה.
1.5. תוצאת הבדיקה
הצגת שם העובד בתיבת טקסט
1.6. בדיקה עברה/נכשלה
הבדיקה עברה בהצלחה

2. Case2

2.1. מטרה
בדיקת הוספת מסמך חדש למאגר המסמכים.
2.2. קלט
הכנסת פרטי המסמך ולחיצה על כפתור "הוסף".
2.3. פלט מצופה (קריטריון הצלחה/כשלון)
הצגת הודעה: "המסמך צורף בהצלחה"
קריטריון לכישלון – הודעת שגיאת SQL.
קריטריון להצלחה – קבלת : "המסמך צורף בהצלחה".
2.4. תהליכי בדיקה
2.4.1. מריצים את סביבת הפיתוח.
2.4.2. נכנסים למערכת.
2.4.3. לוחצים על כפתור "מנהל הידע"
2.4.4. לוחצים על כפתור "הוסף אירוע".
2.4.5. מכניסים את פרטי הנוהל ולוחצים "הוסף".
2.4.6. מוודאים שאין הודעת שגיאה.
2.5. תוצאת הבדיקה
הצגת הודעה: "המסמך צורף בהצלחה"
2.6. בדיקה עברה/נכשלה
הבדיקה עברה בהצלחה

3. Case3

3.1. מטרה
בדיקת הצגת כל האירועים הקיימים במערכת.
3.2. קלט
לחיצת כפתור "הצג כל האירועים".
3.3. פלט מצופה (קריטריון הצלחה/כשלון)
הצגת טבלת האירועים באמצעות DataGridView.
קריטריון לכישלון – הודעת שגיאת SQL.
קריטריון להצלחה – הטבלה מוצגת במלואה.

3.4. תהליכי בדיקה
3.4.1. מריצים את סביבת הפיתוח.
3.4.2. נכנסים למערכת.
3.4.3. לוחצים על כפתור "מנהל האירועים"
3.4.4. לוחצים על כפתור " הצג כל האירועים".
3.4.5. מוודאים שאין הודעת שגיאה.
3.5. תוצאת הבדיקה
הטבלה מוצגת במלואה
3.6. בדיקה עברה/נכשלה
הבדיקה עברה בהצלחה

4. Case4

4.1. מטרה
בדיקת הכנסת קלט תקין בעת ביצוע חיפוש עובד.
4.2. קלט
הכנסת ת"ז עובד ולחיצה על כפתור "חפש".
4.3. פלט מצופה (קריטריון הצלחה/כשלון)
הצגת פרטי העובד באמצעות DataGridView.
קריטריון לכישלון – רקע תיבת הטקסט לאחר סיום הכנסת הנתונים אדומה והודעת שגיאה (הצבע מראה בדיקת קלט דינאמית בהתאם לתבנית הרצויה, במקרה זה מותר להכניס רק ספרות).
קריטריון להצלחה – רקע תיבת הטקסט לאחר סיום הכנסת הנתונים ירוקה והצגת פרטי העובד.
4.4. תהליכי בדיקה
4.4.1. מריצים את סביבת הפיתוח.
4.4.2. נכנסים למערכת.
4.4.3. לוחצים על כפתור "מנהל המשתמשים".
4.4.4. מכניסים את ת"ז העובד בתיבה של פרטי החיפוש.
4.4.5. לוחצים על כפתור "חפש".
4.4.6. מוודאים שאין הודעת שגיאה.
4.5. תוצאת הבדיקה
רקע תיבת הטקסט לאחר סיום הכנסת הנתונים ירוקה והצגת פרטי העובד
4.6. בדיקה עברה/נכשלה
הבדיקה עברה בהצלחה

5. Case5

5.1. מטרה
בדיקת קיום מסמך בבסיס הנתונים לפני מחיקתו.
5.2. קלט
סימון המסמך הרצוי ב DataGridView ולחיצה על כפתור "מחק".
5.3. פלט מצופה (קריטריון הצלחה/כשלון)
הודעת הצלחה: "המסמך נמחק בהצלחה".
קריטריון לכישלון – הודעת שגיאה: "המסמך לא קיים בבסיס הנתונים".
קריטריון להצלחה – הודעת הצלחה ומחיקת המסמך מטבלת המסמכים.
5.4. תהליכי בדיקה
5.4.1. מריצים את סביבת הפיתוח.
5.4.2. נכנסים למערכת.
5.4.3. לוחצים על כפתור "מנהל הידע"
5.4.4. לוחצים על כפתור "הצג כל המסמכים".
5.4.5. מסמנים את המסמך הרצוי.
5.4.6. לוחצים "מחק".
5.4.7. מוודאים שאין הודעת שגיאה.
5.5. תוצאת הבדיקה
הודעת הצלחה ומחיקת המסמך מטבלת המסמכים
5.6. בדיקה עברה/נכשלה
הבדיקה עברה בהצלחה

6. Case6

6.1. מטרה
בדיקת הצגת נתונים האירוע בצורה תקינה.
6.2. קלט
לחיצה כפולה על הרשומה של האירוע הרצוי ב DataGridView.
6.3. פלט מצופה (קריטריון הצלחה/כשלון)
הצגת חלון עם פרטי האירוע.
קריטריון לכישלון – הודעת שגיאה.
קריטריון להצלחה – פרטי האירוע המופיעים החלון זהים לרשומה שנחלצה בטבלה.
6.4. תהליכי בדיקה
6.4.1. מריצים את סביבת הפיתוח.
6.4.2. נכנסים למערכת.
6.4.3. לוחצים על כפתור "מנהל האירועים"
6.4.4. לוחצים על כפתור " הצג כל האירועים".
6.4.5. לוחצים לחיצה כולה על האירוע הרצוי.
6.4.6. מוודאים שנפתח חלון עם פרטי האירוע.
6.5. תוצאת הבדיקה
פרטי האירוע המופיעים החלון זהים לרשומה שנחלצה בטבלה
6.6. בדיקה עברה/נכשלה
הבדיקה עברה בהצלחה

7. Case7

7.1. מטרה
בדיקת תקינות כפתורי "קודם"/"הבא" בחלון פרטי האירוע.
7.2. קלט
לחיצה על כפתור "קודם"/"הבא" בחלון פרטי האירוע.
7.3. פלט מצופה (קריטריון הצלחה/כשלון)
הצגת שם האירוע הבא לפי ה DataGridView במנהל האירועים.
קריטריון לכישלון – הודעת שגיאה (חשוב לציין שברגע שמגיעים לתחילת או לסוף ה DataGridView, מקבלים הודעה מתאימה).
קריטריון להצלחה – קבלת פרטי האירוע הקודם/הבא.
7.4. תהליכי בדיקה
7.4.1. מריצים את סביבת הפיתוח.
7.4.2. נכנסים למערכת.
7.4.3. לוחצים על כפתור "מנהל האירועים".
7.4.4. לוחצים על כפתור "הצג כל האירועים".
7.4.5. לוחצים לחיצה כפולה על אחד הרשומות ומקבלים חלון פרטי האירוע.
7.4.6. לוחצים על כפתור "קודם"/"הבא".
7.4.7. מוודאים שקיבלנו פרטי אירוע נכונים.
7.5. תוצאת הבדיקה
קבלת פרטי האירוע הקודם/הבא
7.6. בדיקה עברה/נכשלה
הבדיקה עברה בהצלחה

8. Case8

8.1. מטרה
בדיקת קבלת נוהל נכון לטיפול באירוע נתון.
8.2. קלט
לחיצה על כפתור "הצג נוהל טיפול" במסך של פרטי האירוע.
8.3. פלט מצופה (קריטריון הצלחה/כשלון)
הצגת רוטינת נוהל טיפול באירוע.
קריטריון לכישלון – הודעת שגיאה: "לא קיים נוהל טיפול עבור סוג אירוע זה".
קריטריון להצלחה – הצגת פרטי נוהל לטיפול באירוע..
8.4. תהליכי בדיקה
8.4.1. מריצים את סביבת הפיתוח.
8.4.2. נכנסים למערכת.
8.4.3. לוחצים על כפתור מנהל האירועים.
8.4.4. לוחצים על כפתור " הצג כל האירועים".
8.4.5. לוחצים לחיצה כפולה על ה DataGridView ומקבלים פרטי אירוע.
8.4.6. לוחצים על כפתור "הצג נוהל טיפול".
8.4.6.1. מוודאים שאין הודעת שגיאה.
8.5. תוצאת הבדיקה
הצגת פרטי נוהל לטיפול באירוע
8.6. בדיקה עברה/נכשלה
הבדיקה עברה בהצלחה

9. Case9

9.1. מטרה
בדיקת תיוק עובד לפריט טיפול.

9.2. קלט
לחיצה על כפתור "התערבות עובד" במסך של טיפול באירוע, הכנס מספר עובד ולחיצת "חפש".

9.3. פלט מצופה (קריטריון הצלחה/כשלון)
הצגת הודעה "העובד עודכן בהצלחה".
קריטריון לכישלון – הודעת שגיאה: "מספר העובד לא נמצא בבסיס הנתונים".
קריטריון להצלחה – הצגת הודעה "העובד עודכן בהצלחה" ולאחר מכן הצגת פרטי העובד.

9.4. תהליכי בדיקה
9.4.1. מריצים את סביבת הפיתוח.
9.4.2. נכנסים למערכת.
9.4.3. לוחצים על כפתור מנהל האירועים.
9.4.4. לוחצים על כפתור "הצג כל האירועים".
9.4.5. לוחצים לחיצה כפולה על ה DataGridView ומקבלים פרטי אירוע.
9.4.6. לוחצים על כפתור "הצג נוהל טיפול".
9.4.7. תוך כדי עדכון הטיפול באירוע לוחצים " Involve worker" ולאחר מכניסים מספר עובד ולוחצים "חפש".
9.4.8. מוודאים שאין הודעת שגיאה.

9.5. תוצאת הבדיקה
הצגת הודעה "העובד עודכן בהצלחה".

9.6. בדיקה עברה/נכשלה
הבדיקה עברה בהצלחה

10. Case10

10.1. מטרה
בדיקת עדכון סטאטוס אירוע לאחר סגירת אירוע

10.2. קלט
סימון כל הפריטים בטיפול באירוע ולחיצת "Apply".

10.3. פלט מצופה (קריטריון הצלחה/כשלון)
הצגת הודעה "סטאטוס האירוע עודכן בהצלחה".
קריטריון לכישלון – הודעת שגיאה: "שגיאת SQL".
קריטריון להצלחה – הצגת הודעה: "סטאטוס האירוע עודכן בהצלחה" ובדיקה פעם נוספת בפרטי האירוע הסגור.

10.4. תהליכי בדיקה
10.4.1. מריצים את סביבת הפיתוח.
10.4.2. נכנסים למערכת.
10.4.3. לוחצים על כפתור מנהל האירועים.
10.4.4. לוחצים על כפתור "הצג כל האירועים".
10.4.5. לוחצים לחיצה כפולה על ה DataGridView ומקבלים פרטי אירוע.
10.4.6. לוחצים על כפתור "הצג נוהל טיפול".
10.4.7. מסמנים את כל פרטי הטיפול בנוהל המוצג ולוחצים "Apply".
10.4.8. מוודאים שאין הודעת שגיאה.

10.5. תוצאת הבדיקה
הצגת הודעה "סטאטוס האירוע עודכן בהצלחה".

10.6. בדיקה עברה/נכשלה
הבדיקה עברה בהצלחה

נספח ה - Software Project Management Plan (SPMP)

1. Introduction

נפסח זה מציג את הצד הניהולי של שלב תכנון ועיצוב המערכת.

בין היתר מסמך זה מכיל את נושא ניהול הסיכונים המפורט בטבלה שכוללת עבור כל סיכוני הפרויקט: קטגוריות סיכון, תיאור הסיכון, סיכוי שסיכון מסוים עלול להתממש, השפעתו של כל סיכון ובמידה ויש גם דרך לפתרון.

נושא חשוב נוסף שמוצג בנספח זה הוא נושא תכנון לוח הזמנים אשר כולל את השינויים שפורטו בדוח ביניים 1 ומיוצג ע"י תוכנת MS – Project.

2. Risk Management

להלן ניתוח סיכונים מעודכן של הפרויקט כאשר מאז תחילת הפרויקט לא היה שום שינוי מבחינת ניתוח והערכת סיכונים:

	קטגוריית הסיכון	תיאור הסיכון	סיכוי הסיכון	השפעת הסיכון	דרך לפתרון הסיכון
1	תכנון ראשוני בלתי מספיק	הערכה לא נכונה של גודל המערכת שיש לפתח	ממוצע (25%-50%)	אי עמידה בלוח הזמנים של הפרויקט (רצינית)	הגדרה מפורשת של היקף הפרויקט לפני תחילתו
2	ניהול פרויקט בצורה לא נכונה	במקום התכנון והאפיון עוברים ישר למימוש	נמוך (10%-25%)	אי שביעות רצון מהמוצר מצד הלקוח ואי עמידה בלוח הזמנים של הפרויקט (קטסטרופלית)	הגדרה ומעקב צמוד אחרי ביצוע הפרויקט כאשר יש צורך להגדיר תנאים במעבר לכל שלב כאשר חלק מאותם תנאים יהיה סיום מלא של ביצוע השלב הקודם.
3	שינויים בדרישות הלקוח	יותר שינויים בדרישות הלקוח מהצפוי	ממוצע (25%-50%)	אי עמידה בלוח הזמנים ואי עמידה בתקציב הפרויקט (רצינית)	הגדרה מסודרת של דרישות הלקוח לפני שמתחילים בבניית הפרויקט

4	גורמים טכנולוגיים	באג מערכת בסביבת פיתוח	נמוך (-10% 25%)	אי עמידה בלוח הזמנים ואי שביעות רצון מהמוצר מצד הלקוח (קטסטרופלית)	לימוד טוב של סביבת הפיתוח כולל כל החסרונות וכאשר רואים בעיה עתידית בתהליך הפיתוח יש צורך לבדוק חלופות מתאימות
5	גורמים טכנולוגיים	מעבר לטכנולוגיה יותר מתקדמת	נמוך (-10% 25%)	אי תאימות המוצר לארגון (קטסטרופלית)	לערוך סקר טכנולוגיות לפני שמתחילים את הפרויקט ולעשות הכנה מוקדמת למקרה שבו יכולות להיכנס לשוק טכנולוגיות חדשות
6	גורמים אנושים	מחלת מפתח הפרויקט	נמוך (-10% 25%)	אי עמידה בלוח הזמנים של הפרויקט (רצינית)	אין אפשרות למנוע
7	גורמים ארגוניים שונים	שינוים במבנה הארגון	נמוך (-10% 25%)	אי תאימות המוצר לארגון או לאי עמידה בלוח הזמנים של הפרויקט (רצינית)	ללמוד טוב את מבנה הארגון עבורו מפתחים את המערכת ובדיקה מקדימה האם הארגון עלול לעבור שינויים אדמיניסטרטיביים

טבלה1 - ניהול הסיכונים הקיימים

3. Evaluation of the SPMP

3.1. הפרויקט מתקדם כמתוכנן כאשר עד עתה בוצעו השלבים הבאים:

3.1.1. שלב א (עד הצעת הפרויקט)

3.1.1.1. סיום הגדרת דרישות לקוח.

3.1.1.2. סיום הגדרות מטרות ומדדי הפרויקט.

3.1.1.3. סיכום כתיבת הצעת פרויקט מורחבת.

3.1.1.4. תיקון הערות מנחה עבור הצעת פרויקט מורחבת.

3.1.2. שלב ב (עד להגשת דו"ח ביניים 1)

3.1.2.1. סיום קריאת חומר תיאורטי.

3.1.2.2. סיום הגדרת אפיון המערכת

3.1.2.3. התחלת עבודה עם כלי פיתוח.

3.1.2.4. סיום כתיבת דו"ח ביניים 1.

3.1.3. שלב ג (עד להגשת דו"ח ביניים 2 ואבטיפוס)

3.1.3.1. התקנת סביבת עבודה.

3.1.3.2. מימוש מודלי המערכת.

3.1.3.3. סיום כתיבת דו"ח ביניים 2 והגשת אבטיפוס של המערכת.

3.1.3.4. תיקון הערות מנחה עבור דו"ח ביניים 2 (במידה ויש הערות).

3.2. מנקודה זו ועד סיום הפרויקט נותרו 2 השלבים הבאים:

3.2.1. שלב ד (עד להגשת פרויקט + הספר)

3.2.1.1. סיום מימוש מודלי המערכת.

3.2.1.2. בדיקות תוכנה עבור מודולי המערכת.

3.2.1.3. סיום בניית המערכת לאחר בדיקות מערכת תקינות.

3.2.1.4. הגשת ספר פרויקט.

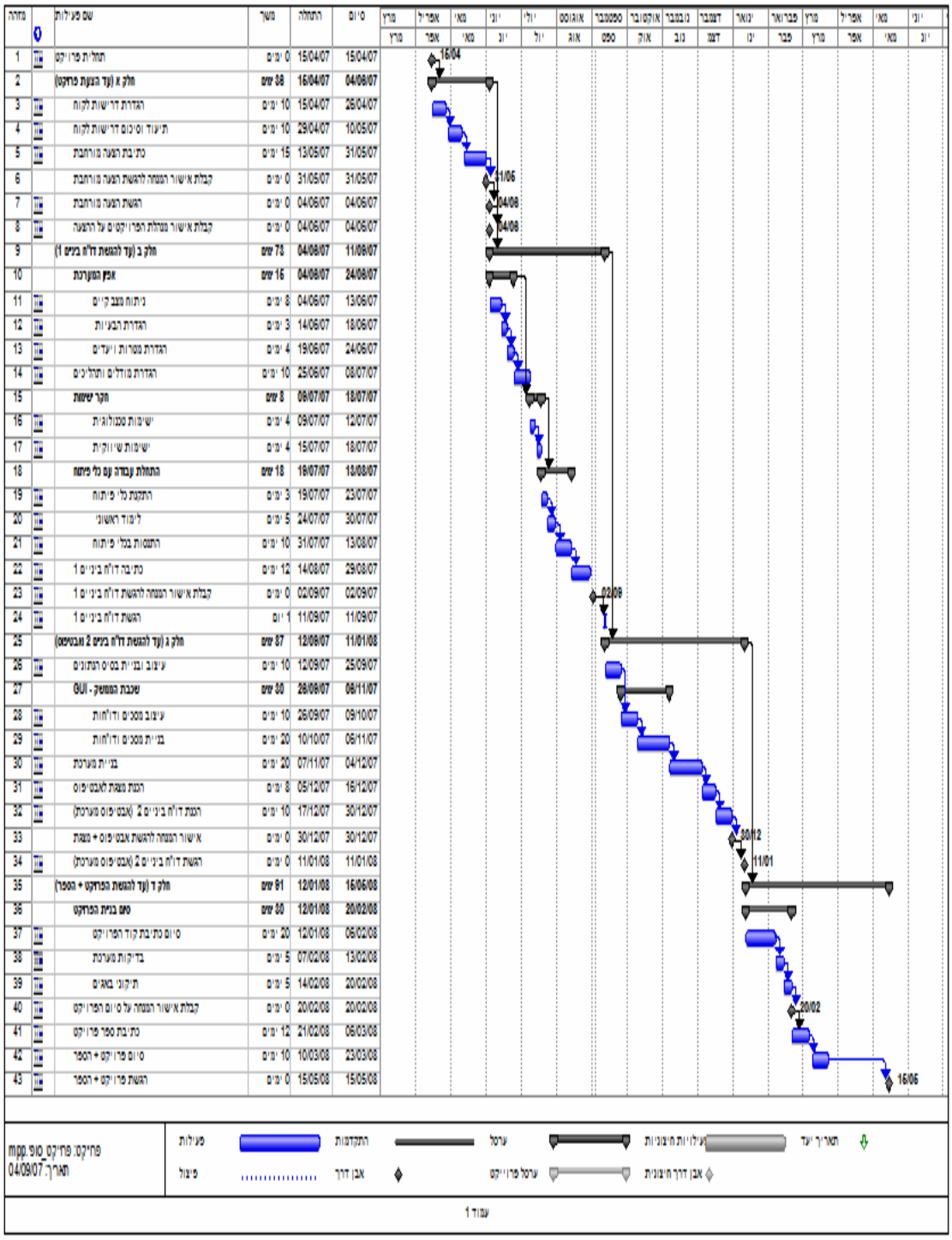
3.2.1.5. תיקון הערות מנחה עבור ספר הפרויקט (במידה ויש הערות).

3.2.1.6. העברת מוצר מוגמר ללקוח (כולל מדריך למשתמש).

3.2.1.7. הצגת מצגת שיווקית עבור המערכת שנבנתה בפרויקט.

4. Schedule

להלן לוח הזמנים המעודכן של הפרויקט:



איור 1 – תרשים משימות לפי לוח זמנים בעזרת תוכנת MS-Project

נספח ו - הצעת פרויקט מורחבת

מחלקת הנדסת תוכנה

שם הפרויקט:

מערכת מידע לניהול, תיעוד ורישום אירועי אבטחת מידע עבור מוקד אבטחת מידע של בנק דיסקונט

הצעת פרויקט

שם הסטודנט: ויקטור קייטמזוב

מספר תעודת זהות: 306859893

שם המנחה: יונית שוורץ וורבר

תאריך הגשה: 21.05.2007

תוכן העניינים

2. תמצית נושא הפרויקט

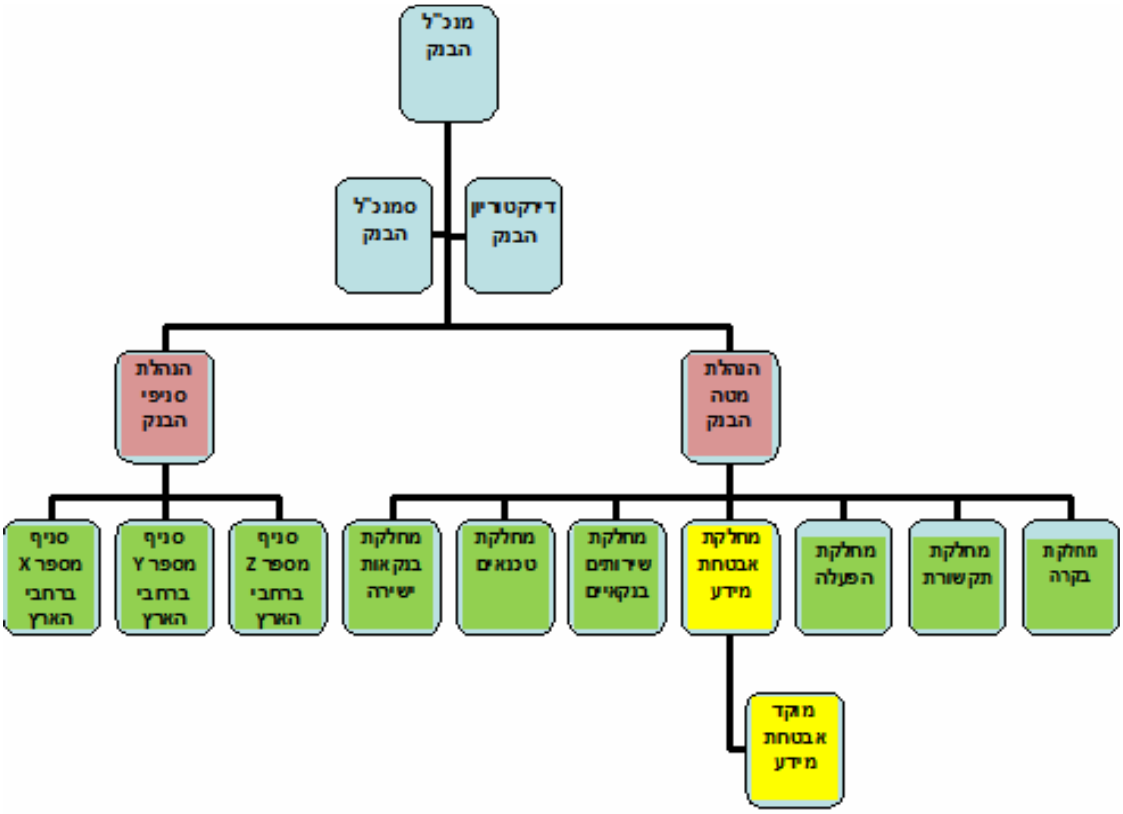
הפרויקט הוא מערכת מידע לניהול, רישום ותיעוד אירועי אבטחת מידע.

המערכת מיועדת למחלקת אבטחת מידע של בנק דיסקונט. כחלק מהעבודה השוטפת קיים במחלקה מוקד אבטחת מידע שתפקידו לנטר את כל אירועי אבטחת המידע של הבנק. המוקד פעיל במשך 24 שעות ביממה, 7 ימים בשבוע ומאויש ע"י בקרי אבטחת מידע שעושים בקרה על מערכות הבנק. במוקד אבטחת מידע קיימות מספר מערכות שונות לניטור אירוע אבטחת מידע (שאותן אין באפשרותי לתאר עקב איסור מוחלט מטעם הבנק לפרסום מידע הקשור למערכות אלה).

כיום בעבודה השוטפת של הבנק בכלל ומחלקת אבטחת מידע בפרט קיימים עשרות סוגים של אירועים חריגים, כאשר לכל אירוע יש להכין דו"ח מיוחד ולהעבירו לגורמים הרלוונטיים בבנק. מערכת המידע שתבנה כחלק מהפרויקט תשמש את המוקד בעבודה השוטפת של תיעוד אירועים והפקת הדו"חות עבור אירועי אבטחת מידע חריגים, כמו כן המערכת תכיל נוהל טיפול עבור כל אירוע חריג שאירע באחת המערכות. בנוסף המערכת תאפשר הפקת דוחות פנימיים של המוקד, כגון: דו"ח חפיפה בחילוף המשמרות בין הבקרים במוקד ודו"ח מרוכז של אירועים חריגים לפי סינון רלוונטי עבור מנהל המוקד. דבר נוסף שיהיה במערכת הוא פורטל לניהול ידע שיכיל את כל הנהלים הרלוונטיים לעבודה השוטפת. חשוב לציין שבשלב הראשוני מערכת המידע שתבנה עבור המוקד לא תתממשק למערכות הבנק השונות אלא תשמש לניהול העבודה השוטפת של המוקד בלבד.

למערכת יהיה בסיס נתונים אחד שיכיל 3 נושאים עיקריים: בסיס נתונים של אירועים שנרשמו במערכת, בסיס נתונים של עובדי הבנק לצורך בדיקת הרשאות מערכת ולצורך יצירת קשר בזמן הטיפול באירועים חריגים ובסיס נתונים של נהלי המוקד ומסמכי מדיניות אבטחת המידע של הבנק עבור הפורטל לניהול הידע.

2.1 תרשים הארגון



איור 1 – תרשים הארגון

2.2 מבנה הארגון

מבנה הארגון מתחלק לשני גורמים מרכזיים: הנהלת מטה בנק והנהלת סניפי הבנק כאשר לכל גורם יש מחלקות משלו ורשת מחשבים משלו. הגורמים שרלוונטיים למערכת הם שני הגורמים המרכזיים (מסומנים בורוד) וחלק מהמחלקות הפנימיות של כל גורם (מסומנים בירוק). כאשר מתחיל טיפול באירוע חריג שנרשם במערכת, ברוב המקרים הגורמים הראשונים שלהם יש לדווח על האירוע החריג הם הגורמים המרכזיים שהם הנהלת מטה הבנק והנהלת סניפי הבנק (בהתאם למקור האירוע) ורק לאחר מכן בהתאם לסוג האירוע, הדיווח מגיע למחלקות הפנימיות לצורך המשך טיפול באירוע. בחלק קטן מהאירועים החריגים אין צורך להודיע לגורמים המרכזיים, אלא רק למחלקות הפנימיות של כל גורם.

2.3 הגורמים המעורבים בהכנת הפרויקט

לקוח מטעם הבנק: מאיר סלר - מנהל מוקד אבטחת מידע, בנק דיסקונט.

מנחת הפרויקט: גב' יונית שוורץ וורבר - פרויקטורית במגמה להנדסת תוכנה.

מפתח הפרויקט: ויקטור קייטמזוב.

3.מטרות ומדדים

3.1 מצב קיים

בנק דיסקונט מנהל את המידע בצורה ממוחשבת באמצעות מערכות מתקדמות שמטפלות בנתוני הלקוחות של הבנק.

מחלקת אבטחת מידע עובדת עם תוכנות מתקדמות שמאפשרות ניטור מדיניות אבטחת מידע של הבנק, כאשר קיימות מספר מערכות שרצות במקביל ובכל מערכת קיימת תוכנה שמנטרת אירועים רלוונטיים המוגדרים בפרמטרים שלה.

הגוף שאחראי על ריכוז האירועים הוא מוקד אבטחת מידע שמאויש ע"י בקרי אבטחת מידע 24 שעות ביממה, 7 ימים בשבוע. חלק מהעבודה השוטפת של בקרי המוקד היא לזהות אירועים חריגים במערכות אבטחת המידע של הבנק ולדווח על כך לגורמים הנדרשים לפי נהלי העבודה של מחלקת אבטחת מידע.

3.2 בעיות - מצב קיים

1. הפקת דו"חות

הפקת הדו"חות של האירועים והעברתם לגורמים הרלוונטיים בבנק לא נעשית בצורה יעילה ומהירה, זאת עקב העובדה שלוקח הרבה זמן לערוך כל דו"ח על סעיפיו השונים, כלומר קיים צורך למלא מחדש את כל שדות הדו"ח הנדרשים על מנת לשמור על הצגת הדו"ח בצורה פורמאלית שתואמת את נהלי הארגון.

2. חפיפה בין משמרות המוקד

בעת חילופי משמרות בין בקרי המוקד, החפיפה על האירועים שהתרחשו במהלך המשמרת נעשית באופן ידני, דבר זה גורם לפעמים להעברת פרטים לא מדויקים ובדרך כלל החפיפות נמשכות זמן רב.

3. חוסר בקרה

אין מערכת שתרכז את כל מגוון האירועים החריגים ביחד ותגדיר את סטאטוס הטיפול בכל אירוע חריג. דבר זה עלול להוביל למצב של התעלמות מאירועים חריגים שעדין בטיפול וטרם הסתיימו.

4. ניהול ידע

אין מערכת מרוכזת לניהול ידע שמגדירה את כל נהלי העבודה השוטפת עבור בקרי המוקד. דבר זה גורם לחוסר מקצועיות שעלול להוביל לטעויות קריטיות מצד הבקרים בזמן הטיפול באירועי אבטחת מידע חריגים.

3.3 מטרות

- ריכוז כל האירועים החריגים מכל המערכות במוקד אבטחת מידע תחת מערכת אחת (פעולה זאת תעשה ע"י בקרי המוקד ולא ע"י התממשקות מערכת המידע למערכות הבנק השונות).
- הגדרת ופיתוח פורטל לניהול ידע עבור בקרי המוקד.
- ארגון מסודר של אירועים חריגים שנרשמו במערכת לפי סוג אירוע ורמת חומרתו.
- ניהול מסודר של דו"חות אירועים חריגים שאירעו במערכות השונות של המוקד.

3.4 מדדים

- צמצום זמן חפיפה בין המשמרות ב - 50% באמצעות דו"ח ממוחשב.
- גישה מהירה תוך 10 שניות לנושא ספציפי במאגר הדו"חות.
- הפקת דו"חות אירועים מרוכזים לפי פרמטרים רלוונטיים בדיוק של 100%.
- דיווח על אירוע תוך 3 דקות מקבלת התראת מערכת ע"י דו"ח מוכן עם פרטי האירוע לכל הגורמים הרלוונטיים של הבנק.
- הצגת נוהל טיפול באירוע תוך 10 שניות מרגע רישום האירוע במערכת.

4.דרישות ותיחום

4.1 דרישות המערכת

דרישות מידע ופונקציונאלית

- המערכת תגדיר תבנית דו"ח מיוחדת לכל סוג של אירוע אבטחת מידע חריג.
- המערכת תאפשר להגדיר סטאטוס עבור כל אירוע ואת אופן הטיפול בו.
- המערכת תכלול בסיס נתונים שיכיל את כל הדו"חות שקיימים במערכת.
- המערכת תכיל פורטל לניהול ידע שירכז את כל הנהלים הרלוונטיים לעבודה השוטפת.

דרישות שימושיות ואנושיות

- המערכת צריכה להיות ידידותית למשתמש ונוחה להפעלה.
- למערכת יהיה מצורף מדריך למשתמש.

דרישות ביצועים ודרישות מבצעיות

- במערכת ניתן יהיה לבצע סינון דו"חות לפי הפרמטרים הנדרשים.
- המערכת תמיין את הדו"חות בהתאם לסוג המערכת שממנה הגיעה התראת אבטחת מידע חריגה.

דרישות בטחון ואבטחה

- לכל משתמש במערכת יהיה שם משתמש וסיסמא ייחודיים.
- לכל משתמש יהיה סוג הרשאה המתאימה לתפקיד.
- חל איסור מוחלט להוציא דו"חות מערכת מחוץ לתחומי העבודה.
- בסיס הנתונים של המערכת יהיה מאובטח וגישה אליו תתאפשר בהתאם לסוג ההרשאה של משתמש המערכת.

דרישות תחזוקה ותמיכה

- אחת לחודש יש צורך לבצע גיבוי של בסיס הנתונים במערכת.

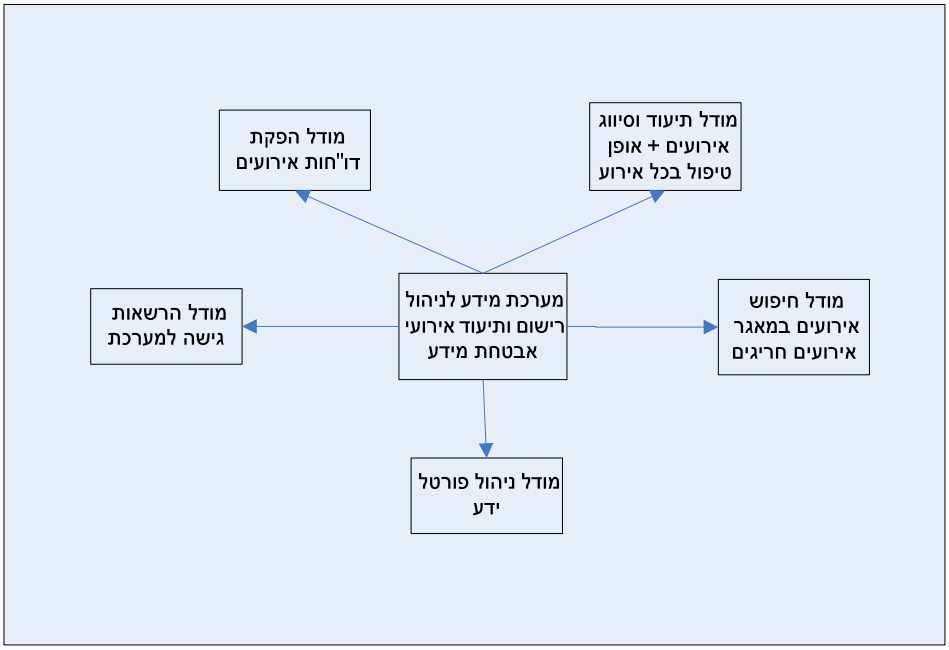
4.2 תיחום העבודה

מערכת מידע.

4.3 תיחום המוצר

תיחום המוצר הוא מערכת מידע לניהול, רישום ותיעוד אירועי אבטחת מידע שתבנה בהתאם לדרישות הלקוח.

4.4 המודלים שיהיו במערכת



איור 2 – מודלי המערכת

המערכת תכלול 5 מודלים עיקריים כאשר לכל מודול תהיה פונקציונאליות שתאפיין אותו. המודולים ידברו ביניהם דרך המערכת המרכזית שתנהל את כל הפונקציונאליות של כל המודולים.

<u>המודולים שיהיו במערכת</u>:

1.1. מודל הרשאות גישה למערכת.

1.2. מודל תיעוד וסיווג אירועים + אופן טיפול בכל אירוע.

1.3. מודל חיפוש אירועים במאגר אירועים חריגים.

1.4. מודל הפקת דו"חות אירועים.

1.5. מודל לניהול פורטל ידע.

5.תוכנית בדיקות ראשונית

סביבת הבדיקות שתבדוק את תפקוד המערכת שנבנתה היא מחשב עם בסיס נתונים מקומי שיריץ את האפליקציה של מערכת המידע , כמו כן ניתן להציג סימולציה של מספר מחשבים שמחוברים ברשת למחשב (שרת) שיכיל את בסיס הנתונים של מערכת המידע.

אין שום צורך לחבר את המחשבים שיריצו את האפליקציה למערכות הבנק, זאת עקב העובדה שהמערכת שתבנה תשמש בשלב הראשוני את מוקד אבטחת המידע בלבד.

כמו כן חשוב לציין שמערכת המידע שתבנה בפרויקט לא תתממשק למערכות הבנק עקב העובדה שדבר זה כרוך באישורי הנהלה רבים, בהשקעה רבה מאוד בבניית המערכת ובצורך לבצע שינויים חלקיים במערכות הקיימות של הבנק.

6.תיאור דרך הביצוע המתוכנן

6.1 שלבים בביצוע הפרויקט

שלב 1: הנדסת המערכת

- שלב זה יכלול את כל הפעולות הנדרשות על מנת לאפיין את המערכת והפרויקט: הגשת הצעת פרויקט, ניתוח המצב הקיים והבעיות שבו. אפיון המערכת יתבצע במהלך הפגישות בין הלקוח למפתח הפרויקט.
- בסוף שלב זה יאוגדו כל המסקנות והניתוחים לדו"ח המסכם הראשון.

שלב 2: ארכיטקטורה

- בשלב זה יעוצבו בפירוט כל המודולים של המערכת.עיצוב המערכת תעשה ע"י מפתח הפרויקט בעזרת לימוד מבנה נכון של מערכות מידע.

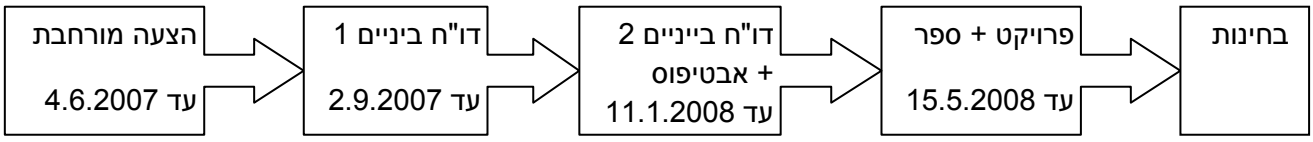
שלב 3: בניית המערכת (אב טיפוס)

- שלב זה יכלול את הפיתוח של המערכת על בסיס ההגדרות של השלבים הקודמים המפורטות בדו"ח הביניים הראשון ובהצעת הפרויקט.
- הפיתוח יהיה פיתוח מונחה עצמים - מימוש בעזרת שפה עלית SQL + C#.NET. server
- בסוף שלב זה יוצג האב טיפוס של המערכת ויוגש דו"ח ביניים שני.

שלב 4: המשך בניית המערכת

- בשלב זה יימשך פיתוח המערכת – המודולים הקיימים ומודולים נוספים שלא היו דרושים לאב טיפוס.
- בסיום הבנייה התיקונים והבדיקות ייכתב ויוגש ספר הפרויקט והפרויקט עצמו.

6.2 מועדי הגשות במשך ביצוע הפרויקט



איור 3 – מועדי הגשות

7.האמצעים/הכלים הנדרשים לביצוע הפרויקט

7.1 סביבת פיתוח

כלים:

- MS-Project.
- UML (Use Case, Class Diagram, Sequence Diagram).
- Microsoft Office – Visio (לצורך תרשימי DFD להגדרת המערכת).

תוכנות:

- מערכת הפעלה מסוג Windows XP Professional.
- תוכנות Microsoft Office עם עדיפות לגרסאות 2003/ 2007.
- Microsoft Visual Studio® .NET 2005 (Framework 2.0)
- Microsoft SQL Server 2005 (עבור בסיס הנתונים).

חומרה:

עבור מחשב שמפתח את המערכת המידע:

- מעבד פנטיום 4.
- זיכרון 512MB ומעלה.

7.2 סביבת ריצה

תוכנות:

- מערכת הפעלה מסוג Windows 2000 ומעלה עם עדיפות למערכת הפעלה Windows XP Professional.
- תוכנות Microsoft Office.
- Microsoft SQL Server 2005 (עבור בסיס הנתונים).

חומרה:

עבור מחשב או רשת מחשבים שמריצות את מערכת המידע:

- מעבד פנטיום 4.
- זיכרון 256MB ומעלה.
- כרטיס רשת.
- כבל רשת.

8.ניתוח חלופות מערכתי ראשוני

8.1 חלופות לרכיבים בפרויקט

קטגוריה	חלופות אפשריות	חלופה מועדפת	הסבר לעדיפות
שפת תכנות	4. C#.NET 5. VB.NET 6. JAVA	C#.NET	נשתמש בשפת C#.NET כי היא שפה הכי נוחה למימוש של מערכת מידע באמצעות GUI. בנוסף זוהי השפה שבה משתמשים במחלקת פיתוח של הבנק לצורך פיתוח אפליקציות נוספות הקיימות בבנק. שימוש בשפה זו יאפשר בעתיד אפשרות התממשקות נוחה וקלה יותר של מערכת המידע למערכות אחרות הקיימות בבנק. שפות VB.NET ו JAVA פחות יתאימו במידה ונרצה להתממשק בעתיד למערכות אחרות בבנק.
סביבת פיתוח	4. Microsoft Visual Studio .NET 5. Eclipse 6. Intellij IDEA	Microsoft Visual Studio® .NET	עבור שפת תכנות C#.NET נשתמש בעורך Microsoft Visual Studio.NET כי זהו העורך המתאים ביותר לפיתוח בשפה זו. שני העורכים האחרים הם עורכי JAVA בעלי פונקציונאליות זהה, ובמידה ונרצה להשתמש בשפה זו, נעדיף את עורך Eclipse עקב העובדה שעורך זה הוא חינמי לצורך פיתוח לעומת Intellij IDEA שהוא בתשלום.
סביבת עבודה	מערכות הפעלה Windows שונות החל מגרסת שנת 2000	Windows XP Professional	נעדיף להשתמש בגרסת מערכת הפעלה כמה שיותר חדשה החל מגרסת Windows XP, אחרי זה Windows 2003 ובסוף Windows 2000 על גרסאותיהם השונות, עקב העובדה שככל שהגרסה יותר חדישה יש לה יותר פונקציונאליות ויותר יכולות התממשקות למערכות השונות.

בסיס נתונים	6. SQL Server 7. Access 8. Oracle 9. MySQL	SQL Server	נשתמש בבסיס נתונים SQL Server כי זה בסיס הנתונים שמובנה בעורך Microsoft Visual Studio.NET. בסיסי נתונים אחרים כגון Access ו Oracle שלא מובנים בעורך, מתאימים לניהול בסיס נתונים של מערכות מידע, כאשר Oracle בהשוואה ל Access שמיועד לסביבת עבודה קטנה או בינונית, יכול לעבוד גם בסביבת עבודה גדולה. לא נשתמש ב MySQL כי הוא יותר מתאים לנהל בסיס נתונים של אפליקציות אינטרנט (שממומשות באמצעות שפת PHP).

טבלה 1 - ניתוח חלופות לרכיבים בפרויקט

8.2 חלופות פרויקט (מערכות דומות עם אותה פונקציונאליות)

1. חברת Ness - מערכת לניהול אבטחת מידע מרכזי

מערכת שדואגת להקמת מערכי Security Information Management, מכילה מתודולוגיה בניהול אירועי אבטחת מידע עם דגש על שילוב פתרונות קיימים בארגון (שליטה ובקרה, ניהול אירועים).

2. חברת CA - מערכת לניהול מידע אבטחתי

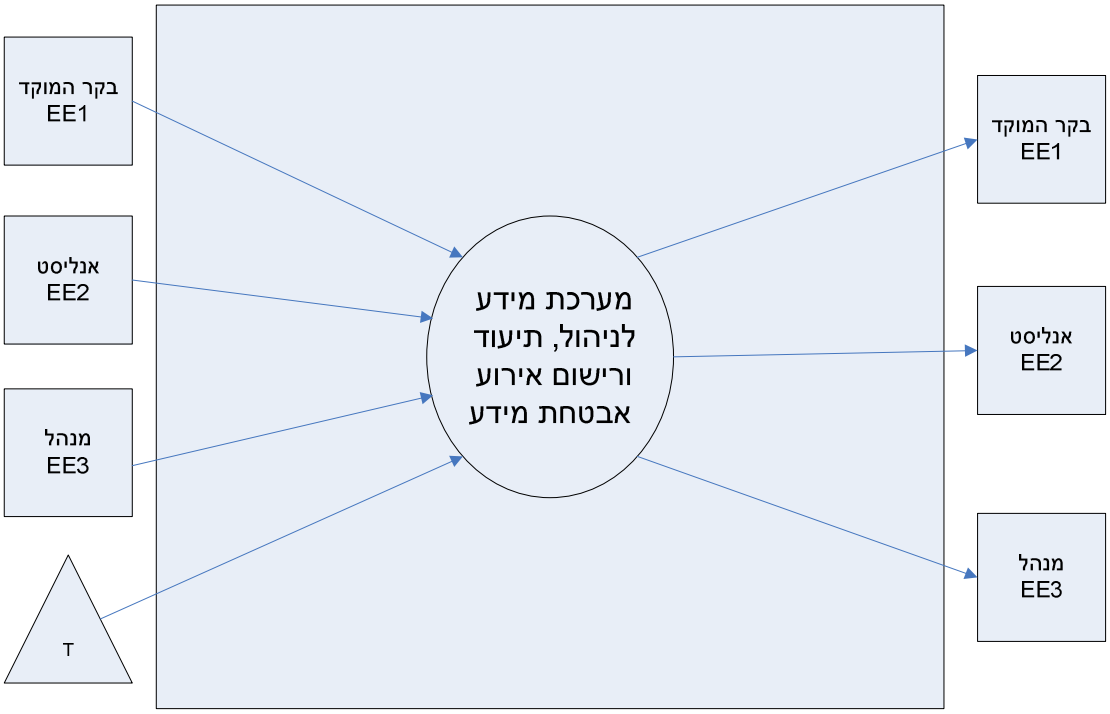
המערכת יודעת כיצד לשלוט בשטף אירועי האבטחה ולהבין מה חשוב ומה קריטי לתהליכים העסקיים של הארגון.

3. פתרון נוסף שיכול לשמש כחלופה לפרויקט הוא פיתוח תוכנה דומה עם אותה הפונקציונאליות ע"י מחלקת פיתוח של הבנק.

הערה: השוואה נרחבת של החלופות תימסר בשלב דו"ח ביניים 1.

9.ניתוח פונקציונאלי ראשוני

9.1 תרשים הקשר (Context Diagram)

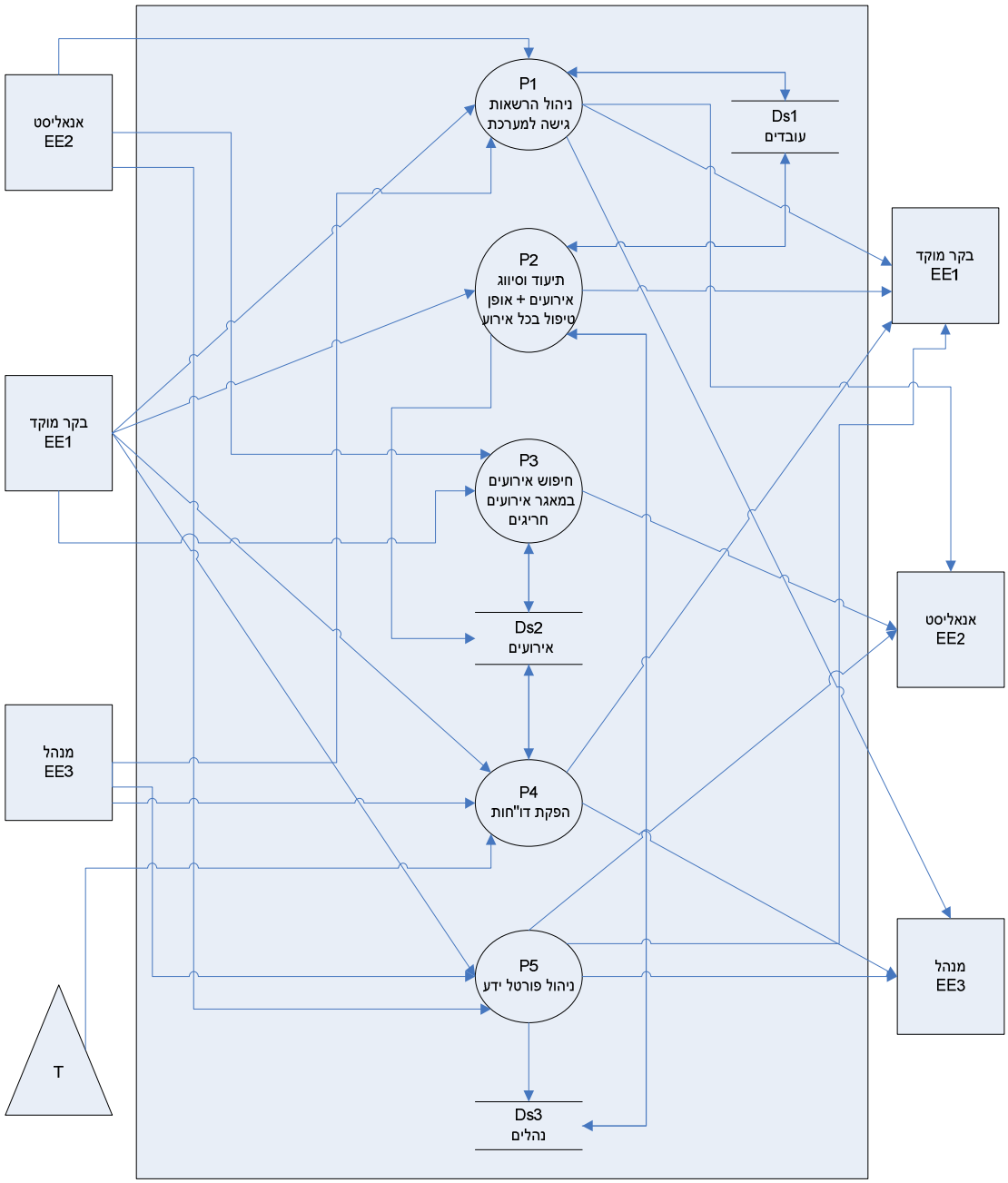


איור 4 - תרשים הקשר

מערכת המידע שתפותח בפרויקט תקבל נתונים מבקרי המוקד, אנליסטים או ממנהל המוקד. כמו כן פעם בחודש המערכת תפיק דו"חות חודשיים לפי הגדרת פרמטרים מתאימים של מפתח המערכת לפי דרישות מנהל המוקד. מערכת המידע תעביר נתונים לבקרי המוקד, אנליסטים או למנהל המוקד כפי שמתואר באיור 4.

9.2 תרשים פונקציונאלי DFD-0

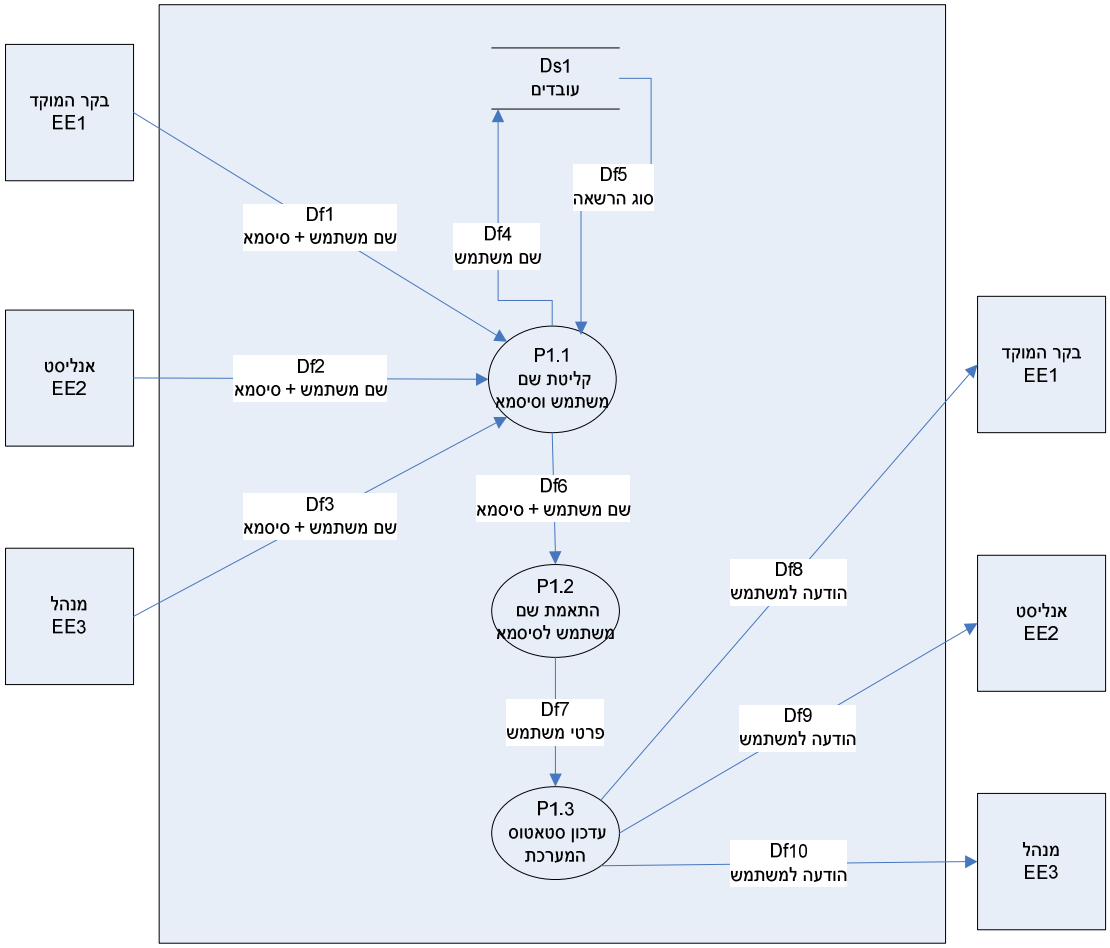
DFD 0 - מערכת לניהול, תיעוד ורישום אירועי אבטחת מידע



איור 5 - תרשים DFD-0 של המערכת

9.3 פירוט תהליכים DFD-1

DFD 1 של P1 - ניהול הרשאות גישה למערכת

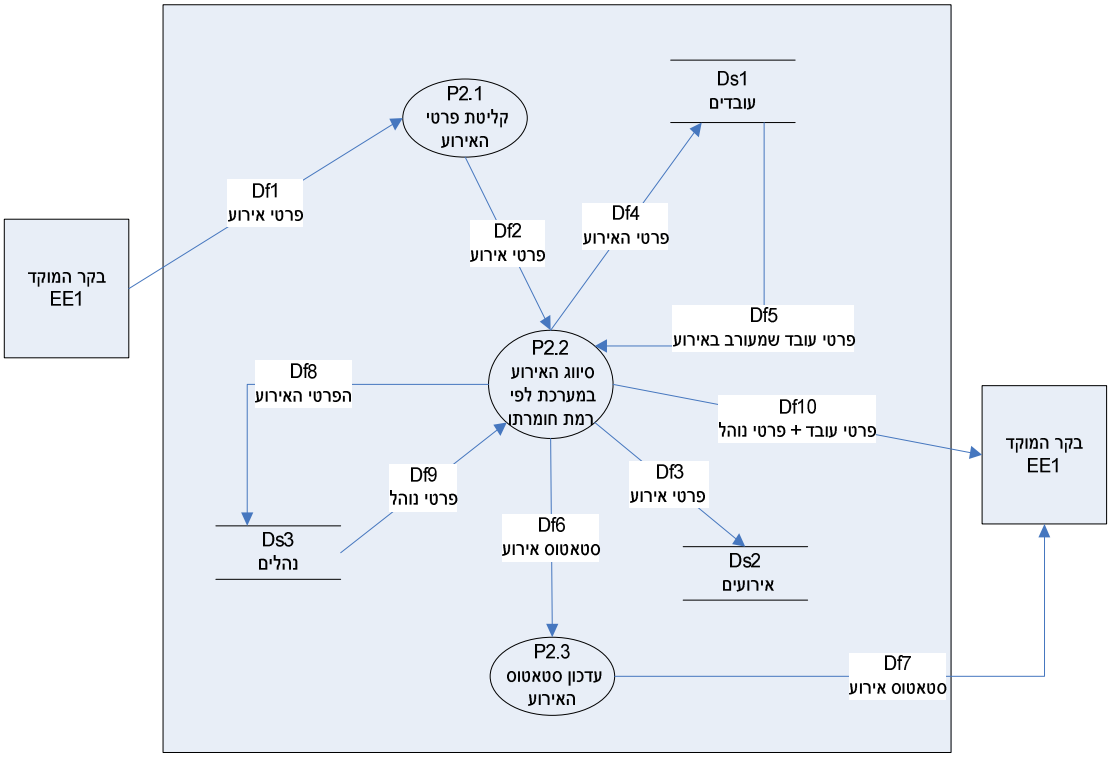


איור 5.1 – תרשים תהליך ניהול הרשאות גישה

P1 - תהליך ניהול הרשאות גישה למערכת (איור 5.1)

תהליך זה יקלוט את שם המשתמש והסיסמא ויבדוק מול בסיס הנתונים של עובדי הבנק האם העובד מורשה לגשת למערכת.

DFD 1 של P2 - תהליך תיעוד וסיווג אירועים + אופן הטיפול בהם

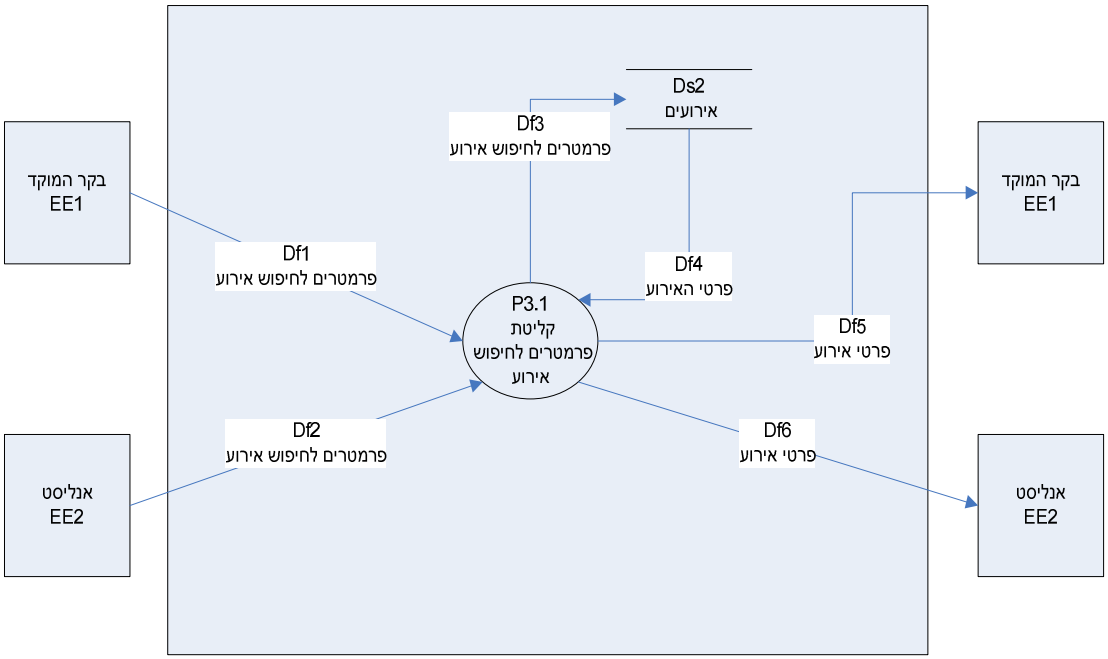


איור 5.2 – תרשים תהליך תיעוד וסיווג אירועים + אופן הטיפול בכל אירוע

P2 - תהליך תיעוד וסיווג אירועים + אופן הטיפול בכל אירוע (איור 5.2)

תהליך זה יגדיר במערכת את סוגי האירועים השונים (כאשר לכל סוג אירוע תהיה מחיצה נפרדת) ואת רמת החומרה של כל אירוע, כמו כן לאחר רישום האירוע בבסיס הנתונים, המערכת תפנה את הבקר להמשך תפעול האירוע בהתאם לנוהלי העבודה של מוקד אבטחת המידע.

DFD 1 של P3 - תהליך חיפוש אירועים במאגר אירועים חריגים

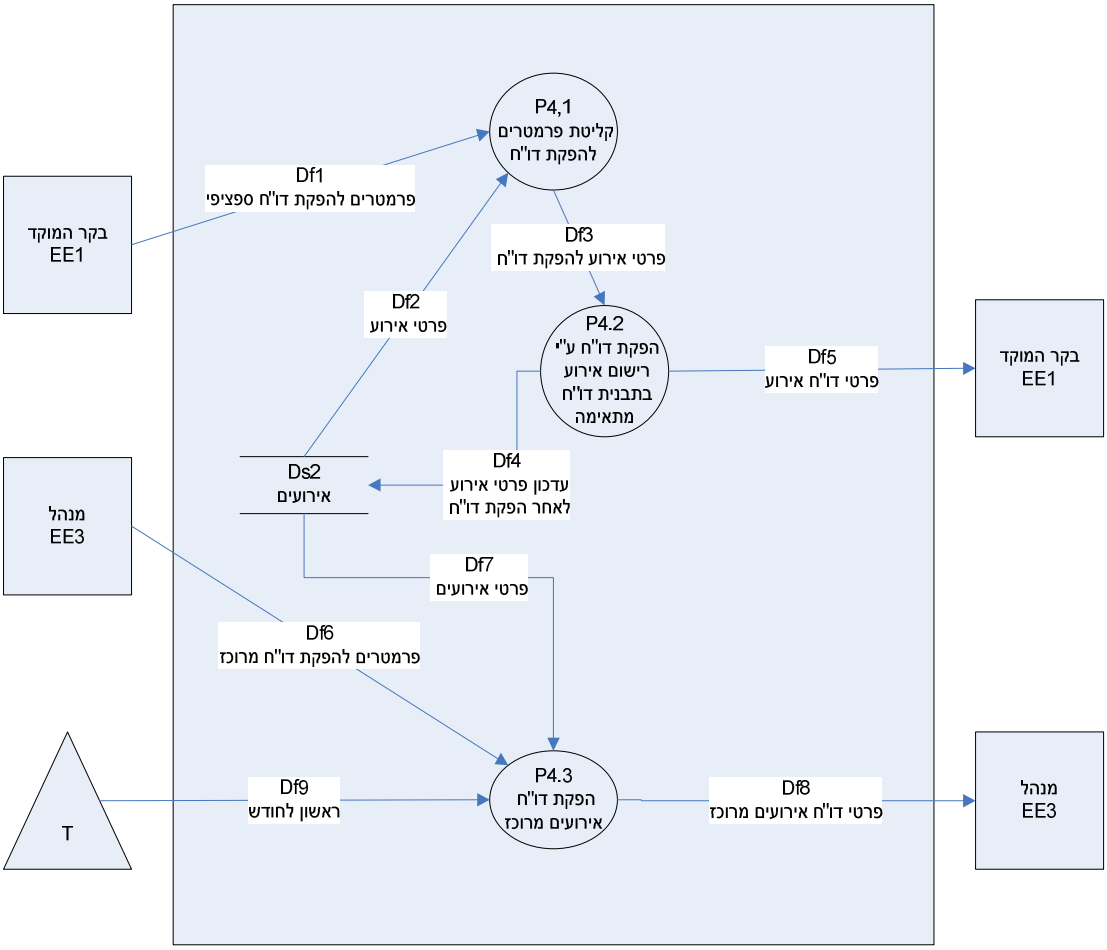


איור 5.3 – תרשים תהליך חיפוש אירועים במאגר מידע

P3 - תהליך חיפוש אירועים במאגר אירועים חריגים (איור 5.3)

תהליך זה יאפשר לחפש אירועים לפי פרמטרים של רמת סיווג שבהן האירועים הוגדרו במערכת, החיפוש יעשה מתוך מאגר אירועים שנרשמו במערכת.

DFD 1 של P4 - תהליך הפקת דו"חות

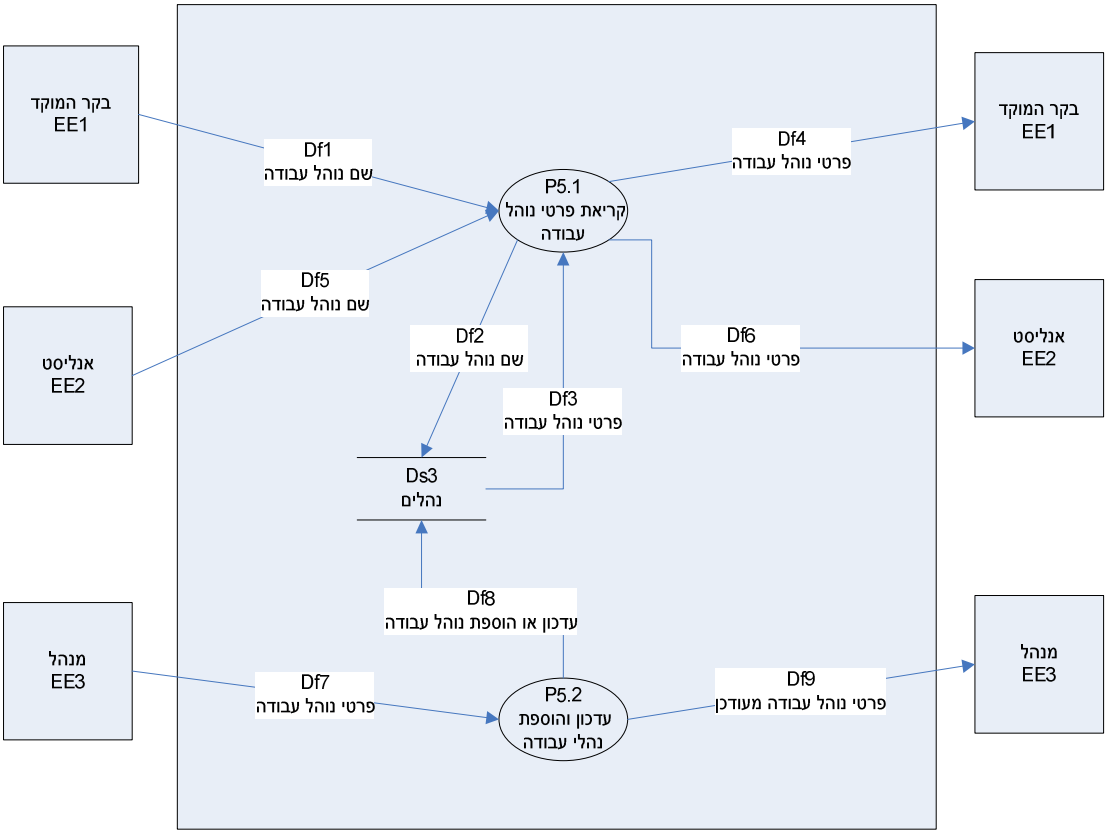


איור 5.4 – תרשים תהליך הפקת דו"חות חריגים ודו"חות חפיפה בין המשמרות

P4 - תהליך הפקת דו"חות (5.4)

תהליך זה יפיק דו"חות אירועים ע"י רישום פרטי אירועים שנלקחו מבסיס הנתונים בתבניות דו"ח מתאימות. בנוסף להפקת דו"חות של אירועים חריגים, יופקו גם דו"חות של משימות שוטפות הכלולות בדו"ח חפיפה בין משמרות שבו ירוכזו כלל האירועים שהתרחשו בזמן המשמרת. כמו כן המנהל יוכל להפיק דו"ח אירועים מרוכז לפי סוגי אירועים למיניהם.

DFD 1 של P5 - תהליך לניהול פורטל ידע



איור 5.5 – תרשים תהליך לניהול פורטל הידע

P5 - תהליך לניהול פורטל הידע (איור 5.5)

תהליך זה יכיל קישור מתוך המערכת לפורטל ידע שבו ירוכזו כל נהלי העבודה של המוקד וחומר לימודי אודות האפליקציות השונות שבהם המוקד משתמש בעבודה השוטפת.

9.4 פירוט מאגרי בסיס הנתונים

Ds1 - בסיס נתונים של עובדי הבנק לצורך בדיקת הרשאות מערכת ולצורך יצירת קשר בזמן הטיפול באירועים חריגים.

Ds2 - בסיס נתונים של אירועים שנרשמו במערכת.

Ds3 - בסיס נתונים עבור נהלי המוקד ומסמכי מדיניות אבטחת המידע של הבנק עבור הפורטל לניהול הידע.

10.פערי ידע שעל הסטודנט להשלים

- קישור טפסי מערכת המידע לבסיס הנתונים SQL – Server, פער זה יושלם בחודש הקרוב במהלך הקורס: טכנולוגיות NET.
- ארגון נכון של מבנה מערכת מידע באמצעות חלוקת המערכת למודולים השונים וקישור בין המודולים הקיימים במערכת, פער זה יושלם בעזת למידה עצמית לאחר סיום הסמסטר הנוכחי.
- עיצוב ומימוש יעיל של מערכת המידע תוך מיצוי מקסימאלי של הטכנולוגיות שעומדות לרשותי בעורך (Framework 2.0) NET2005. Microsoft Visual Studio®, פער זה יושלם בחודש הקרוב במהלך הקורס: טכנולוגיות NET.

11.ניהול סיכונים

	קטגוריית הסיכון	תיאור הסיכון	סיכוי הסיכון	השפעת הסיכון	דרך לפתרון הסיכון
1	תכנון ראשוני בלתי מספיק	הערכה לא נכונה של גודל המערכת שיש לפתח	ממוצע (25%-50%)	אי עמידה בלוח הזמנים של הפרויקט (רצינית)	הגדרה מפורשת של היקף הפרויקט לפני תחילתו
2	ניהול פרויקט בצורה לא נכונה	במקום התכנון והאפיון עוברים ישר למימוש	נמוך (10%-25%)	אי שביעות רצון מהמוצר מצד הלקוח ואי עמידה בלוח הזמנים של הפרויקט (קטסטרופלית)	הגדרה ומעקב צמוד אחרי ביצוע הפרויקט כאשר יש צורך להגדיר תנאים במעבר לכל שלב כאשר חלק מאותם תנאים יהיה סיום מלא של ביצוע השלב הקודם.
3	שינויים בדרישות הלקוח	יותר שינויים בדרישות הלקוח מהצפוי	ממוצע (25%-50%)	אי עמידה בלוח הזמנים ואי עמידה בתקציב הפרויקט (רצינית)	הגדרה מסודרת של דרישות הלקוח לפני שמתחילים בבניית הפרויקט
4	גורמים טכנולוגיים	באג מערכת בסביבת פיתוח	נמוך (10%-25%)	אי עמידה בלוח הזמנים ואי שביעות רצון מהמוצר מצד הלקוח (קטסטרופלית)	לימוד טוב של סביבת הפיתוח כולל כל החסרונות וכאשר רואים בעיה עתידית בתהליך הפיתוח יש צורך לבדוק חלופות מתאימות
5	גורמים טכנולוגיים	מעבר לטכנולוגיה יותר מתקדמת	נמוך (10%-25%)	אי תאימות המוצר לארגון (קטסטרופלית)	לערוך סקר טכנולוגיות לפני שמתחילים את הפרויקט ולעשות הכנה מוקדמת למקרה שבו יכולות להיכנס לשוק טכנולוגיות חדשות
6	גורמים אנושים	מחלת מפתח הפרויקט	נמוך (10%-25%)	אי עמידה בלוח הזמנים של הפרויקט (רצינית)	אין אפשרות למנוע
7	גורמים ארגוניים שונים	שינוים במבנה הארגון	נמוך (10%-25%)	אי תאימות המוצר לארגון או לאי עמידה בלוח הזמנים של הפרויקט (רצינית)	ללמוד טוב את מבנה הארגון עבורו מפתחים את המערכת ובדיקה מקדימה האם הארגון עלול לעבור שינויים אדמיניסטרטיביים

טבלה 2 - ניתוח ראשוני של הסיכונים הקיימים

12.תוצרי הפרויקט

1. הצעת פרויקט מורחבת (מסמך זה).
2. דו"ח ביניים ראשון.
3. דו"ח ביניים שני.
4. אבטיפוס.
5. ספר פרויקט.
6. המערכת עצמה + מדריך כללי למשתמשי המערכת.
7. מצגת שיווקית.

13.פירוק המערכת ליחידות עבודה (WBS)

1. **שלב א (עד הצעת הפרויקט)**

1.1 סיום הגדרת דרישות לקוח.

1.2 סיום הגדרות מטרות ומדדי הפרויקט.

1.3 סיכום כתיבת הצעת פרויקט מורחבת.

1.4 תיקון הערות מנחה עבור הצעת פרויקט מורחבת (במידה ויש הערות).

2. **שלב ב (עד להגשת דו"ח ביניים 1)**

2.1 סיום קריאת חומר תיאורטי.

2.2 סיום הגדרת אפיון המערכת

2.3 התחלת עבודה עם כלי פיתוח.

2.4 סיום כתיבת דו"ח ביניים 1.

2.5 תיקון הערות מנחה עבור דו"ח ביניים 1 (במידה ויש הערות).

3. **שלב ג (עד להגשת דו"ח ביניים 2 ואבטיפוס)**

3.1 התקנת סביבת עבודה.

3.2 מימוש מודלי המערכת.

3.3 סיום כתיבת דו"ח ביניים 2 והגשת אבטיפוס של המערכת.

3.4 תיקון הערות מנחה עבור דו"ח ביניים 2 (במידה ויש הערות).

4. **שלב ד (עד להגשת פרויקט + הספר)**

4.1 סיום מימוש מודלי המערכת.

4.2 בדיקות תוכנה עבור מודולי המערכת.

4.3 סיום בניית המערכת לאחר בדיקות מערכת תקינות.

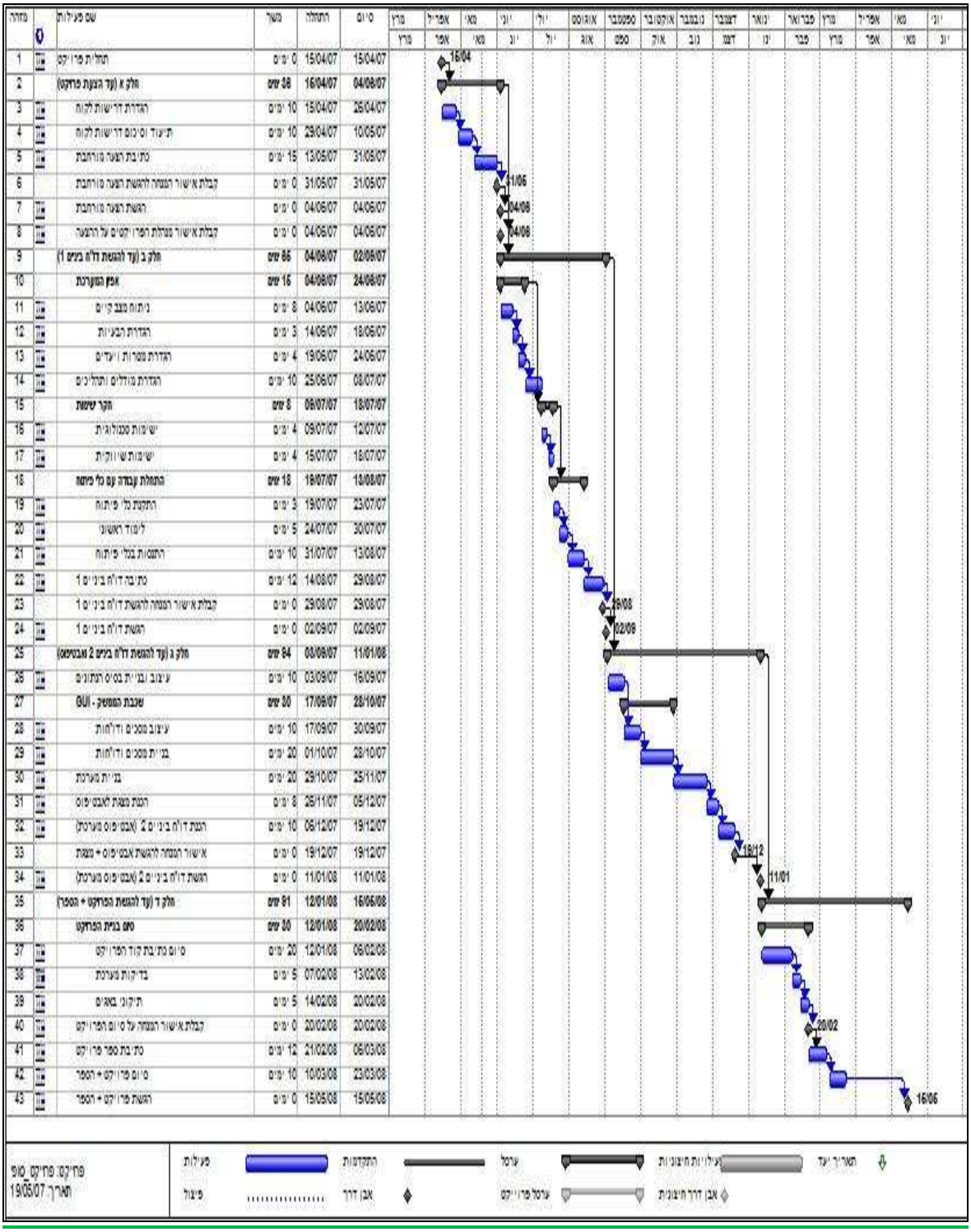
4.4 הגשת ספר פרויקט.

4.5 תיקון הערות מנחה עבור ספר הפרויקט (במידה ויש הערות).

4.6 העברת מוצר מוגמר ללקוח (כולל מדריך למשתמש).

4.7 הצגת מצגת שיווקית עבור המערכת שנבנתה בפרויקט.

14. תוכנית עבודה של הפרויקט



איור 6 – תרשים משימות לפי לוח זמנים בעזרת תוכנת MS-Project

15.רשימת מקורות

1. ירושלמי, אורי. (2003).***.NET & #C - מדריך מקצועי***. מרכז הדרכה 2000.
2. פרץ, שובל.(1998).***תכנון,ניתוח ועיצוב מערכות מידע, כרך א - תכנון***. תל אביב: האוניברסיטה הפתוחה.
3. פרץ, שובל.(1998).***תכנון,ניתוח ועיצוב מערכות מידע, כרך ב - ניתוח ועיצוב***. תל אביב: האוניברסיטה הפתוחה.
4. פרץ, שובל.(1998).***תכנון,ניתוח ועיצוב מערכות מידע, כרך ג - אב- טיפוס, כלי פיתוח, יישום וגישת העצמים***. תל אביב: האוניברסיטה הפתוחה.
5. Jones, Bradley L. (2002). ***#C - סדנת לימוד***.הרצליה: הוד- עמי.
6. Sharp, John.(2006).***Visual C# 2005 Step by Step***. Redmond, Washington: Microsoft Press.

נספח ז - ניהול בנקאי תקין, הוראת בנק ישראל 357 - ניהול טכנולוגיית המידע

1. עיקרי ההוראה:

1.1. פיקוח וניהול

1.2. הערכת סיכונים

1.3. אבטחת מידע

1.4. גיבוי והתאוששות

1.5. מיקור חוץ

1.6. שירותי בנקאות בתקשורת

2. הסבר על כל פרק בהוראה 357:

2.1. פיקוח וניהול:

- דיון תקופתי בדירקטוריון ובהנהלה בנושא ניהול טכנולוגיית המידע.
- קביעת נהלים מפורטים לכל תהליך בתפעול, אבטחה, גיבוי ובקרה של טכנולוגיית המידע.
- תיעוד מתאים ועדכני למערך טכנולוגיית המידע.
- קיום נתיב ביקורת ממוכן מבוסס LOG של עצם הגישה, פעולות ושאילתות המבוצעות במערכות המידע.
- יחידה ארגונית לביקורת טכנולוגיית המידע במסגרת הביקורת הפנימית.

2.2. הערכת סיכונים:

ביצוע הערכת סיכונים, שתתעדכן באופן שוטף ומתמיד למזעור אפשרות פגיעה במערך טכנולוגיית המידע.

2.3. אבטחת מידע:

- יישום אמצעי אבטחה למניעה, גילוי, תיקון ותיעוד של חשיפות במערך טכנולוגיית המידע.
- ביצוע סקר בטיחות וניסיונות חדירה מבוקרים.
- זיהוי אישי וחד ערכי של כל גורם בעל גישה למערכות הטכנולוגיות.
- הצפנה של נתונים.
- קישוריות התאגיד לאינטרנט בהתאם לנדרש.

2.4. גיבוי והתאוששות:

- דיון בהנהלה בעקרונות הגיבוי וההתאוששות תוך התייחסות להערכת הסיכונים.
- קיום תכנית מפורטת להפעלת מערך טכנולוגיית המידע במקרים של תקלות ואסונות.

2.5. מיקור חוץ:

- הגדרת קווים מנחים להתקשרות עם גורמים חיצוניים לביצוע פעילויות ניהול, עיבוד ואחסון מידע.
- קבלת הסכמה מראש של המפקח על הבנקים במקרים שונים של מיקור חוץ ובפרט במיקור חוץ של מערכות ליבה ואחסון מידע לגבי לקוחות התאגיד במערכות שאינן בשליטתו הבלעדית.

2.6. שרותי בנקאות בתקשורת:

- הגדרת המושג "בנקאות בתקשורת".
- דירוג רמות שירות של שירותי בנקאות בתקשורת.
- קיום אמצעי זיהוי אישיים וחד ערכיים.
- ניהול סיסמאות.
- קיום אמצעי בקרה בבנקאות בתקשורת.
- הגברת הבקרה לעסקאות בתקשורת לטובת צד שלישי,ניהול רשימת מוטבים ופעילויות בדואר אלקטרוני.